ADAPTIVE POWER MANAGEMENT FOR CYBER-PHYSICAL SYSTEM

A Dissertation Proposal
Presented to
The Academic Faculty

By

Ruizhi Chai

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
School of Electrical and Computer Engineering

Georgia Institute of Technology

May 2019

ADAPTIVE POWER MANAGEMENT FOR CYBER-PHYSICAL SYSTEM

Approved by:

Dr. Ying Zhang, Advisor
School of Electrical and Computer Engineering
Georgia Institute of Technology

Dr. Mary Ann Weitnauer
School of Electrical and Computer Engineering
Georgia Institute of Technology

Dr. Xiaoli Ma
School of Electrical and Computer Engineering
Georgia Institute of Technology

Dr. Ghasson AlRegib
School of Electrical and Computer Engineering
Georgia Institute of Technology

Dr. Yang Wang
School of Civil and Environmental Engineering
Georgia Institute of Technology

Date Approved: January 10, 2019

To my family and all those supported me during this journey.

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to my Ph.D. advisor Dr. Ying Zhang for her guidance during my research and support for my graduate study. Her insight about research, inspiring ideas and kind encouragement has helped me a lot through my graduate study and finally helping me to finish this thesis. Without her support, I couldn't have achieved this degree.

I am fortunate to have many collaborators and co-authors to work with. They include my lab colleagues: Wen Sun, Yen-Pang Lai, Ayca Ermis, Xinhai Pan. In particular, I am grateful to have Wen Sun, who helped us to build the experiment platform and even continuing the support after she left the lab to pursue her dream in graduate school. I also want to thank Yen-Pang Lai, who is really pleasant to work with. We build the experiment system together and I learned a lot from him in the process. I would also like to thank my colleagues in the lab, which include Xiangmin Wei, Zhenhua Xie, Hengzhao Yang, Qianao Ju and Zongyang Xia. It is a great pleasure to work with you during these years.

Moreover, I would like to thank the faculty members that served in my Ph.D. dissertation committee: Dr. Mary Ann Weitnauer, Dr. Xiaoli Ma, Dr. Ghasson AlRegib, Dr. Yang Wang and Dr. Maysam Ghovanloo. Your insightful suggestions and comments really helped me improve my research and helped me to get where I am.

At last, I am deeply grateful for the love and support from my parents and my elder sister. You are the most precious thing I have in this world and I am really sorry for not being able to spend more time with you.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1
INTRODUCTION AND BACKGROUND

The cyber-physical system has three core components. First is the computational platforms, which consists of sensors, actuators, and computers. Second is the communication network that provides the mechanisms for the computer platforms to communicate with each other as well as the end user. Third is the physical plant, which represents the part of system that is not realized with computers or communication networks. It can include mechanical parts, biological or chemical processes, or human operators. The power management of the cyber-physical system has to be designed by taking these three core components into consideration.

The objective of this dissertation is to investigate the power management techniques for cyber-physical systems based on the model predictive control (MPC) method, which has the benefit of handling the complex constraints of cyber-physical systems explicitly. The performance of the MPC is known to be closely related to the accuracy of the prediction model. If it is difficult to model certain part of the cyber-physical system, the power management techniques can be difficult to design. It is therefore critical to build an accurate prediction model for the dynamics of the cyber-physical system. The methods that can be used to obtain such models can be divided into two main categories. One modeling techniques is based on the physics of the system. The other is the black box based modeling techniques.

The first technique is demonstrated through a typical cyber-physical system: wireless sensor network. More specifically, a wireless sensor network uses a supercapacitor as the energy storage device of each node and takes advantage of ambient energy to support long operation time. A network that is designed to detect events with the radar sensor at each node is used as an example for the demonstration. A power management method based

on the MPC is proposed to guarantee a satisfactory event detection probability. The event detection probability of the network is decomposed as the quality of service of each sensor node. The developed method tracks a reference quality of service while satisfying the operation constraints of each node. For each sensor node, the terminal voltage of supercapacitor is closely related to the performance of the radar sensor and it also determines the operation lifetime. However, the terminal behavior of supercapacitor can be difficult to model especially when the charging/discharging power varies in a wide range. In the proposed power management method, a state of charge online prediction method is developed to accurately predict the terminal behavior of the supercapacitor and achieve the full potential of the stored energy. The online prediction method is based on a supercapacitor equivalent circuit model that emulates both the ohmic leakage and charge redistribution phenomena of supercapacitor, and updates the model parameters in a receding horizon fashion according to the most recent charging/discharging history. Since the supercapacitor charging/discharging characteristics is known to be affected by various factors, by incorporating this modeling method into the adaptive power management framework, the performance of the controller could be more robust in practical applications.

Then the black box based modeling technique is demonstrated with the power management of bioimplants. These systems are implanted into the human body to perform the designed operations. The direct coupling between the electronics and biological tissue poses new challenge for the development of power management techniques. One such challenge is the overheating caused by the device operation. It is reported that a temperature increase greater than $1\ ^oC$ in the brain tissue could cause long term damage. Therefore, it is vital to take the heat generation into consideration when designing the power management techniques. In this work, an adaptive power management framework is developed for bioimplants, which optimize the application performance while respecting the system constraints, especially the temperature constraints. More specifically, the power management framework is developed based on the MPC method to optimize the device operation

while maintaining a safe operation temperature. A simplified thermal model is developed to provide online temperature prediction and support real-time control. The model parameters are updated online using a proposed recursive multi-step prediction error minimization method (RMSPEM). To improve the robustness of the RMSPEM and support longer prediction interval, a Bayesian recursive multi-step prediction method is proposed, which takes the prior information of the parameters into consideration. More generally, the developed method can also be applied in many real applications where the system model is uncertain or time varying. The power management framework incorporates this adaptive thermal model and is tested via a COMSOL Multiphysics model and a experiment test vehicle that are built to emulate the temperature effect of an implanted Utah electrode array (UEA) in the brain. Simulation and experiment results indicate that the proposed method achieves longer operation time and maintains safe operating temperature of the implanted device.

Through these two applications, the adaptive MPC-based power management framework for cyber-physical systems is investigated. The supercapacitor-powered sensor network case demonstrates a general approach to apply MPC-based power management to a sensor network if the network performance can be decomposed into the quality of service of each sensor node. In addition, a physics-based model, which is preferred if it is suitable for real-time implementation, is developed to predict the state of charge of supercapacitor for supporting the MPC-based adaptive power management. Then, the power management of bioimplant case is used to demonstrate that a black box modeling technique can be used for supporting MPC-based power management if the system model is uncertain or too complicated for real time implementation. Furthermore, a black box modeling method with Bayesian parameter estimation techniques is proposed to improve both the prediction accuracy and robustness.

CHAPTER 2
LITERATURE SURVEY

2.1 Cyber-Physical Systems

Cyber-physical system (CPS) is an integration of computational platform, networking and physical processes. The physical plant is the "physical" part of the system, which can include mechanical parts, biological or chemical processes, or human operators. The "cyber" part of the system monitor and control the physical processes via the onboard sensors, actuators, and processors in the feedback fashion. On the other hand, change of physical processes also affect the operation status of the "cyber" part. Examples of CPS include smart grid [1], robotics systems [2], medical monitoring and stimulation systems [3], and process control systems [4].

The power management of cyber-physical system can be challenging due to the complex interaction between the information infrastructure and the physical process. In this work, the model predictive control (MPC) [5] framework is investigated for power management of cyber-physical systems, since MPC can explicitly handle the complex constraints of cyber-physical systems and is robust to system uncertainty.

2.2 Model Predictive Control

With the rapid development of embedded system and increasing demand for better performance, the theory and technology of MPC has been greatly advanced during the last decade. MPC have achieved great success in the areas of advanced manufacturing, energy, environment, aerospace, automobile, etc for solving constrained optimization control problems, such as energy efficient building control [6], flight control [7], satellite attitude control [8], supply chain management in semiconductor manufacturing [9], and autonomous

vehicle [10].

In this work, MPC is investigated for power management of two cyber-physical systems: a supercapacitor-powered wireless sensor network and a bioimplant. These two systems are selected to demonstrate two major modeling methods typically employed in MPC: the physics-based modeling method and the black box based modeling method, since prediction models with high accuracies play a crucial role in the MPC framework [11, 12, 13].

2.3 Power Management of Sensor Network Systems

The wireless sensor network has been employed in applications like earthquake sensing, weather monitoring or wildlife monitoring [14], radiation monitoring [15], military surveillance [16]. However, the limited lifetime of wireless sensor networks has prevented them from being widely adopted by various long-term applications. Two directions are being investigated to overcome this limitation: developing energy-efficient power management algorithms; and harvesting ambient energy to power wireless sensors. Ambient energy harvesting has the potential to provide infinite amount of energy to the node. The energy harvested from ambient environment, such as solar energy [17], is often stored in a storage device for future use.

Two types of commonly used storage devices are rechargeable batteries [18] and supercapacitors [19]. Supercapacitors are known to have a much longer operational lifetime, simpler charging requirements, fast charge/discharge characteristics and higher power densities, allowing them to supply large currents efficiently [20]. While supercapacitors do feature energy densities that are orders of magnitude lower than batteries, energy harvesting sensor nodes don't have the need to store large amount of energy. Therefore, supercapacitors have been considered as energy buffers in energy harvesting sensor nodes to significantly extend their lifetime. Various supercapacitor models [21, 22] have been developed to support real-time power management. It is shown that in order to make full

use of the stored energy, power management policy needs to be designed based on the supercapacitor state. In this thesis, a new supcapacitor model is proposed to predict the supercapacitor state of charge online. The model can be implemented on a sensor node and support the real-time power management. The model parameters are updated in a receding horizon fashion according to the most recent charging/discharging history.

Other than the energy storage device, sensor nodes are equipped with various sensors to monitor the area of interest. Surveillance systems have used infrared, acoustic, and magnetic sensors for passive sensing, and optical and ultrasonic sensors for active sensing. Radar based sensors have also become an emerging solution with the radar system becoming more efficient and compact. Compared with low-rate sensors such as the temperature sensors and humidity sensors, radar based sensors usually generate raw data at hundreds of kilobits or tens of megabits per second [23]. A radar sensor network is used as an example to demonstrate the power management framework developed in this work.

Due to the limited bandwidth of wireless network, it is important to reduce the communication overhead by in-network data processing. At the node level, the raw data of radar sensor needs to be pre-processed before sending over the communication link. Therefore, it is important to balance the in-network processing and communication. Sensor nodes often have limited energy capacity. In order to extend the operation lifetime, many sensor networks are designed to conserve energy. However, this could leads to reduced system performance. For example, the critical event won't be detected if the radar sensor is turned off when the event happens. Moreover, for radar based sensors, it is also known that the sensor footprint area may be shrinking with the decline of the energy level [24]. In this cases, traditional energy conservation techniques often fail to provide satisfactory performance. It is therefore critical to balance the sensing and energy conservation to guarantee satisfactory system performance.

The aforementioned challenges have brought forward new requirements for the power management of radar based sensor networks. Many research efforts have focused on de-

veloping energy saving methods for sensor networks. Pantazis *et al.* [25] presented a comprehensive survey of the passive and active power conserving mechanisms in wireless sensor networks. Anatasi *et al.* [26] summarized the energy conservation approaches for wireless sensor networks with a focus on reducing the power consumption on the sensor level. The existing research works typically assume the energy consumption of the communication is much higher than the energy consumption due to sensing or processing. Many radio sensor network applications, however, have demonstrated that the power consumption of sensing is comparable to that of radio transmission. Jaleel *et al.* [24, 27] modeled the decaying sensor footprint and developed a duty cycle scheduling scheme to keep the event detection probability of the radar sensor networks at a satisfactory level. However, the proposed methods are not applicable for more complicated nonlinear storage models, like supercapacitor. Moreover, for general radar sensor network applications, communication and in-network processing are also important factors that needs to be investigated when developing power management methods. Moser *et al.* proposed an adaptive power management framework based on the multiparametric programming techniques, which has the benefit of explicit handling of system constratins. The proposed method is developed to account for the unreliable nature of environmental energy and optimize the system performance while respecting the energy neutral operation condition. However, the developed controller use a linear battery model and a linear application performance model which can't solve more complicated cases with nonlinear dynamics involved.

An adaptive power management method for supercapacitor powered systems is developed in this thesis to optimize the system operation while respecting the energy limitation and system constraints. In addition, the online state of charge prediction method is developed to support the MPC based power management framework, which is designed to maintain a satisfactory event detection probability.

2.4 Power Management of Bioimplants

The most challenging problem in bioelectronics are related to biomedical applications, particularly the direct coupling of electronic devices with living organisms. Research work has been conducted for some medical applications such as implantable prosthetic devices [28, 29] and implantable telemetry devices [30, 31]. For example, neuroprosthetics, also known as neural prosthetics, are implantable devices that can restore and rehabilitate sensory and motor functions, such as hearing, vision, limb movement, that might have been damaged as a result of an injury or a disease.

In these applications, the most important issue is the interface between living tissues and the artificial man-made implantable devices. For example, the temperature increase caused by the operation of electronics must also be taken into consideration due to its potential damage to the neighboring living tissues. As the implantable devices become more powerful with the incorporation of high-density, functional electronic components and as the number of stimulation units increases, heating of the surrounding tissues of the implantable device has started to raise safety concern.

Exposing the surrounding tissue of an implantable device to a temperature above the safe limits can cause irreversible damage in the long run. In [32] and [33], it is reported that a patient with an implanted deep brain stimulator (DBS) suffered significant brain damage after diathermy treatment, and subsequently died. Postmortem examinations indicated that the tissue near the lead electrodes of the DBS deteriorated due to overheating. The author of [34] showed that a temperature increase of more than $3\ ^{\circ}$C above normal body temperature will lead to physiological abnormalities such as angiogenesis or necrosis. And when the cerebral cortex temperature of rats increase by $3.4\ ^{\circ}$C, cortical spreading depression was observed [35]. When temperature of guinea pig olfactory cortical slices increase by $2\ ^{\circ}$C, aberrant activity began according to [36]. In [37], it is shown that a temperature increase greater than $1\ ^{\circ}$C may have long-term damage to the brain tissue.

Many factors may contribute to the temperature increase during the operation of an implantable device, which includes the power dissipation of the stimulating electrodes, the power dissipation of the control circuits, and the heat absorption caused by wireless power transmission. The impact of implantable electronics and electromagnetic field on temperature increase can be described based on the Pennes bioheat equation [38]. Lazzi *et al.* [39] investigated the thermal effect of a dual-unit retinal prosthesis, which is used to restore partial vision to the blind. The potential causes of temperature increase, including the electromagnetic fields, are summarized. Moreover, Kim *et al.* [40] investigates the thermal influence of the integrated 3-D Utah electrode array (UEA) device implanted in the brain by numerical simulation and experiment measurement. The experiment validated the numerical model, which shows that the temperature increases linearly with power dissipation of UEA. These research works provide a through investigation on the thermal effect of the implantable device. However, there have been few studies addressing the dynamic thermal management problem of the implantable device, except that a primitive method is implemented in [41] by disabling LED operation when a temperature threshold is reached.

Herein, we investigate the dynamic power management problem of implantable device, with a focus on neural prosthesis. This particular example is educative since it includes most of the potential causes of thermal dissipation: a microchip that could dissipate relatively large power, a telemetry system, and a potentially large number of stimulating electrodes. For this device, consuming as low power as possible is of importance from two standpoints. First, power budget for the operation of implantable microsystems are limited, whether they are battery operated or telemetry operated. Secondly, from a biological point of view, the heat generated by the circuits that dissipate rather large power can potentially cause damage to the neighboring living tissues. However, there exists a trade-off between the performance of the device and its power consumption.

In this work, a real-time adaptive power management framework is developed for implantable devices, in which a simplified thermal model is adopted to support real-time

control and the model parameters are updated online using a proposed model estimation method. The system performance is optimized with the proposed framework while respecting the operation constraints, such as maintaining a safe operating temperature.

2.5 Problems and Challenges

As described above, the power management of cyber-physical system could be difficult to design if a part of the system is difficult to model. A model of a CPS comprises models of physical processes as well as models of the software, computation platforms, and networks. In the example of supercapacitor powered sensor network, the terminal behavior of supercapacitor can be difficult to model especially when the input power varies in a wide range. Since the system only operates when the supercapacitor voltage is above the threshold of interface circuit, and the application performance depends on the energy stored in the supercapacitor, the power management method must be developed based on an accurate supercapacitor model to achieve the full potential of the stored energy. In the example of neural prothesis, the impact of power dissipation on the temperature rise of surrounding tissue must be accurately modeled to design the power management method that ensures the implantable device doesn't overheat too much. This poses new challenges to the traditional power management methods, where the system prediction model is relatively easy to built.

CHAPTER 3
ADAPTIVE POWER MANAGEMENT FOR CONTROLLING EVENT DETECTION PROBABILITY OF RADAR SENSOR NETWORK

3.1 Introduction

In this chapter, the power management of supercapacitor powered radar sensor network is investigated. In this example, the terminal voltage of supercapacitor is a critical factor in that it determines the operation time of the sensor node and also impact the performance of the radar sensor. Therefore, it is important to accurately predict the terminal behavior of supercapacitor, which is achieved by a proposed physics based supercapacitor model. Moreover, an adaptive power management method for supercapacitor powered systems is developed to maintain a satisfactory event detection probability while satisfying the energy limitation and system constraints.

3.2 System Description and Model

Let the radar sensor network be randomly deployed within a domain $\mathcal{D} \in \mathbb{R}^2$ such that the location of each sensor is independent of all the other sensors' locations. The sensors' locations can be modeled as a spatial Poisson point process [42] with constant intensity λ. The network operates in a finite time horizon consisting of discrete time slots $t \in \{0, 1, 2, \ldots, t_{end}\}$, $t_{end} < \infty$ and is used to detect events in $\mathcal{D}$. Each radar sensor has a considerable processing power to perform the following five operations: collecting raw sensor readings, performing pre-processing on the data in the data queue, transmitting or receiving data packets over wireless links and performing online calculation to determine the optimal operation status. The supercapacitor is employed as the energy storage device of each sensor node, since the power consumption of radar sensor is relatively high and

the sensor node can achieve a much longer operation time when equipped with energy harvesting devices.

3.2.1 Energy Storage Model

In this work, a supercapacitor is employed as the energy storage device. Compared to rechargeable batteries, such devices have the benefit of very long charging-discharging cycles and are therefore able to achieve much longer operation time when being used with energy harvesting devices [19]. Additional advantages are high power density, simple charging requirement and robustness to temperature changes, shock, and vibration. However, the supercapacitor is also known to have a lower energy density. Moreover, the supercapacitor has more complex terminal behavior. Besides the ohmic leakage phenomenon, charge redistribution of supercapacitor could also have a significant impact on the terminal behavior. Thus, to achieve the full potential of the stored energy, an energy model that takes both ohmic leakage and charge redistribution into consideration is essential. The two branch equivalent circuit model [22] takes both self-discharge and charge redistribution into consideration. Based on it, the model proposed by Chai *et al.* [43] can be used to predict the terminal behavior of supercapacitor under certain conditions. In Section 3.46, an online state of charge prediction method will be presented, which is able to generate an accurate prediction of the terminal behavior under different loads and estimate the quickly available energy (E_{QA}) stored in the supercapacitor. Due to the high power density of supercapacitor, we use E_{QA} to approximate the available power of the supercapacitor in this paper.

3.2.2 Sensing and Communication Model

At each time slot t, each node collects raw sensor data readings at a rate of $r_s(t) \geq 0$. The maximum sensing rate is represented as r_{max}. We denote the per packet energy cost for sensing operation as $p_s(t)$.

For radar or RF type sensors, the size of sensor footprint is proportional to the available

power of the sensor node. The footprint of sensor node i located at x_i can be represented as a close ball centered at x_i with a radius of $r(t)$. The footprint of a sensor is the region in which an event can be detected. The event is detected with a boolean model, i.e. events are only detected if they are in the footprint and the sensor is turned on. It is shown that if the sensor range model is based on the RF power density function for an isotropic antenna, the sensor footprint is proportional to the available power of the sensor node [44], i.e.

$$r^2(t) = \zeta P(t), \tag{3.1}$$

where ζ is the coefficient of proportionality. Hence, the area of the sensor footprint at time t is

$$A(t) = \pi r(t)^2 = \alpha P(t), \tag{3.2}$$

where $\alpha = \zeta\pi$ is a constant. According to Hsin *et al.* [45], if the sensors with footprint A are randomly deployed with uniform distribution within a large region, the probability of an event being detected can be represented as

$$P_d = 1 - \exp(-\lambda\pi Aq), \tag{3.3}$$

as the number of sensors goes to infinity. In (3.3), q represents the expected value of sensor being on.

The communication scheduling within a network is conducted by computing the contention free link with maximum aggregated weights

$$\mu^*(t) = \arg\max_{\mu(t)\in\mathbb{L}^f} \sum_{(x,y)\in\mu(t)} w_{(x,y)}(t), \tag{3.4}$$

where $\mathbb{L} \in 2^{\mathcal{L}}$ is the set of all contention-free links and $w_{(x,y)}$ is the weight of contention-free link (x, y). For general interference relations, the optimal solution of the scheduling

problem is centralized and NP-hard, which is therefore intractable in practice. Fully distributed suboptimal scheduler is often used to solve the problem in practice. For example, the lightweight Longest Queue First (LQF) has the potential to achieve a near-optimal performance in practice. After the communication links are scheduled, the data forwarding rate $f_{x,y}(t)$ over each link $(x, y) \in \mathcal{L}$ is set between 0 and link capacity $c_{x,y}(t)$, i.e. the maximum number of data packets that can be successfully transmitted from x to y during slot t.

In the problem formulation, this process is simplified by assuming that all data come from a single virtual node and all data are sent to another single virtual node. Thus the details of communication scheduling are abstracted. Moreover, it is assumed that the data rate of sending $f_t(t)$ and receiving $f_r(t)$ can be controlled. This simplifies the details of data forwarding mechanism. In a real sensor network, where there are multiple senders and receivers, the method developed in this dissertation can be used to calculate the average number of data packets to be sent or received after the communication link is scheduled. At the next time slot, when the communication link is rescheduled, the link capacity $c_r(t)$ and $c_t(t)$ can be modified accordingly, and the power management method can be applied again with the updated constraints. The node throughputs $c_r(t)$ and $c_t(t)$ generally don't have abrupt changes, therefore the method developed in this dissertation can be utilized to obtain better performance.

Denote $p_t(t)$ and $p_r(t)$ as the energy prices for transmission and reception at time slot t respectively. For a successful transmission over node i at time slot t, the total energy costs for transmitting and receiving are $f_t(t)p_t(t)$ and $f_r(t)p_r(t)$ respectively.

3.2.3 Data Processing Model

For the radar sensor, each scan produces tens of Megabytes of raw data, which is much higher than the bandwidth available to each radar sensor in a multi-hop mesh network [23]. Therefore, the raw data must be preprocessed prior to transmission. After preprocessing,

the amount of data packets being reduced is represented as the data change rate. It is important to build a model that describes the relation between the data change rate and the energy consumption caused by data processing.

Each node x maintains a data queue $\mathcal{Q}_x(t)$ to store its own scanned data, the data packets preprocessed by itself, and the data packets received from its neighbors, assuming each data packet in the network has a unique identifier that identifies its source ID, data attributes, etc. Let $Q_x(t) \geq 0$ be the length of queue $\mathcal{Q}_x(t)$. Due to the limited RAM size of the sensor node, the data queue $\mathcal{Q}_x(t)$ has a limited size, which can be represented as Q_{max}.

The data change rate is represented as $f_{dp}(t)$, and the maximum amount of data that can be reduced is denoted as $c_{dp}(t)$. The energy consumption of data processing $EC_{dp}(t)$ can be represented as a function of the data change rate as $EC_{dp}(t) = g(f_{dp}(t))$.

Previous work [46] employs a simple averaging technique to down-sample data, through which, neighboring readings are averaged and replaced by their mean. The larger the number of neighboring readings over which the mean is computed, the greater the data change rate. Simple processing operations such as aggregations that are required to compute the maximal, minimal, and average data values normally result in a linear $g(\cdot)$. Let e_1 and e_2 be the energy costs of the atomic operations respectively, then the energy consumption of average filtering can be easily obtained:

$$EC_{dp}(t) = e_1 f_{dp}(t) + e_2. \tag{3.5}$$

The function $g(\cdot)$ could also be a nonlinear function for more complex processing operations such as Kalman-filter based data fusion, and compression algorithms like SPIHT [23].

3.3 Probability of Event Detection

Consider a domain $\mathcal{D}$ where the nodes are randomly deployed. The node deployment can be modelled as a stationary spatial Poisson point process with constant intensity λ. Given a sub space in $\mathcal{D}$ with area A, the probability of having n nodes in this area can be calculated as

$$P_n = \frac{(\lambda A)^n e^{-\lambda A}}{n!}. \tag{3.6}$$

Assume that the total number of nodes in $\mathcal{D}$ is very large and all the nodes are radar based. The footprint of each node i is a closed ball of radius $r_i(t)$, centered at the position of the node x_i. The union of the footprints of all the nodes form the germ-grain model of stochastic geometry. The nodes are on with a probability q.

Assume that a non-persistent event occurs at location $x_e \in \mathcal{D}$. For the event to be detected, it should happen within the footprint of at least one *on* node. If the network is non-decaying, i.e. the node footprints and probabilities of *on* do not change with time. The probability of a non-persistent event not being detected can be represented as

$$P_{un} = \exp(-\lambda A q). \tag{3.7}$$

With a decaying network, the energy of the nodes is consumed when they are on, resulting in a decrease in the radar footprints. This decrease is proportional to the decay in energy. The area of footprint of a node at time t can be represented as (3.2). According to Jaleel *et al.* [27], the probability of an event being detected by a decaying network is given by

$$P_d(t) = 1 - \exp(-\lambda \hat{A}(t) q(t)), \tag{3.8}$$

in which, $\hat{A}(t)$ is the expected coverage of all the nodes.

In order to break down the event detection probability of the network, the quality of service of each node is first defined.

Definition The quality of service (QoS) of node i is defined as

$$1 - \exp(-\lambda A_i(k) q_i(k)). \tag{3.9}$$

Then the average quality of service of all nodes is shown to be the lower bound of the event detection probability of the network.

Theorem 3.3.1 *The event detection probability of a decaying network with M nodes is greater than or equal to*

$$\frac{1}{M} \sum_{i,j=1}^{M} \left(1 - \exp(-\lambda A_i(k) q_j(k))\right) \tag{3.10}$$

as M goes to infinity.

Proof For a decaying network, consider all the nodes of footprint area $A_i(k)$ and sensing ratio $q_j(k)$. Define the ratio of such nodes $\delta_{ij}(k)$ as

$$\delta_{ij}(k) = \frac{\begin{array}{c}\text{number of sensors with footprint} \\ \text{area } A_i(k) \text{ and sensing ratio } q_j(k)\end{array}}{\text{total number of sensors}}. \tag{3.11}$$

Let $N(k)$ be the total number of combinations of footprint $A_i(k)$ and sensing ratio $q_j(k)$ at time k. So $\sum_{i,j=1}^{N(k)} \delta_{ij}(k) = 1$ and $\delta_{ij}(k)\lambda$ is the intensity of nodes with footprint area $A_i(k)$ and sensing ratio $q_j(k)$.

The probability of having n nodes with footprint area $A_i(k)$ and sensing ratio $q_j(k)$ in a given set with area $A_i(k)$ is

$$P_n^{ij}(k) = \frac{(\delta_{ij}(k)\lambda A_i(k))^n \exp(-\delta_{ij}(k)\lambda A_i(k))}{n!}. \tag{3.12}$$

The probability of an event going undetected by all the nodes of footprint area $A_i(k)$

and sensing ratio $q_j(k)$ can be calculated as

$$\begin{aligned} P_u^{ij}(k) &= \sum_{n=0}^{\infty}(1-q_j(k))^n \frac{(\delta_{ij}(k)\lambda A_i(k))^n \exp(-\delta_{ij}(k)\lambda A_i(k))}{n!} \\ &= \exp(-\delta_{ij}(k)\lambda A_i(k)) \sum_{n=0}^{\infty} \frac{[(1-q_j(k))\delta_{ij}(k)\lambda A_i(k)]^n}{n!} \\ &= \exp(-\delta_{ij}(k)\lambda A_i(k)) \exp((1-q_j(k))\delta_{ij}(k)\lambda A_i(k)) \\ &= \exp(-q_j(k)\delta_{ij}(k)\lambda A_i(k)). \end{aligned} \tag{3.13}$$

The total probability of an event going undetected by all the nodes is

$$P_u(k) = \prod_{i,j=1}^{N(k)} P_u^{ij}(k) = \exp(-\lambda[\sum_{i,j=1}^{N(k)} q_j(k)\delta_{ij}(k)A_i(k)]). \tag{3.14}$$

Since

$$\sum_{i,j=1}^{N(k)} q_j(k)\delta_{ij}(k)A_i(k) = \frac{1}{M}\sum_{i,j=1}^{M} q_j(k)A_i(k) \to \overline{(qA)}, \tag{3.15}$$

we have

$$P_u(k) = \exp(-\frac{\lambda}{M}\sum_{i,j=1}^{M} q_j(k)A_i(k)). \tag{3.16}$$

Therefore, the event detection probability of the network is

$$1 - P_u(k) = 1 - \exp(-\frac{\lambda}{M}\sum_{i,j=1}^{M} q_j(k)A_i(k)). \tag{3.17}$$

The theorem is equivalent to

$$1 - \exp(-\frac{\lambda}{M}\sum_{i,j=1}^{M} q_j(k)A_i(k)) \geq \frac{1}{M}\sum_{i,j=1}^{M}(1-\exp(-\lambda A_i(k)q_j(k))). \tag{3.18}$$

To prove (3.18), we have to prove

$$\exp(-\frac{\lambda}{M}\sum_{i,j=1}^{M} q_j(k)A_i(k)) \leq \frac{1}{M}\sum_{i,j=1}^{M}\exp(-\lambda A_i(k)q_j(k)). \tag{3.19}$$

Let

$$f(x) = e^{-\lambda x}. \tag{3.20}$$

$f(x)$ is convex function, since

$$f'' = (\lambda)^2 e^{-\lambda x} > 0. \tag{3.21}$$

Then according to Jensen's inequality, we have

$$\exp(-\lambda \frac{\frac{1}{M}\sum_{i,j=1}^{M} A_i(k)q_j(k)}{\sum_{i,j=1}^{M}\frac{1}{M}}) \leq \frac{\frac{1}{M}\sum_{i,j=1}^{M}\exp(-\lambda A_i(k)q_j(k))}{\sum_{i,j=1}^{M}\frac{1}{M}}, \tag{3.22}$$

which is an equivalent form of (3.19). This completes our proof.

With the Theorem 3.3.1, the event detection probability of the network can be decoupled as the quality of service of each single node.

3.4 Supercapacitor online state of charge prediction

3.4.1 Supercapacitor Physics

Supercapacitors are constructed from two carbon-based electrodes, electrolyte, and a separator [47]. The two carbon-based electrodes are immersed in electrolyte and separated by porous insulating membrane. During the charging process, an electrochemical double-layer of charge is formed at the interface between the electrode and electrolyte. Like conventional capacitors, supercapacitors store charge electrostatically, so there is no charge transfer between electrode and electrolyte. But compared with the conventional electrodes, the porous nature of supercapacitor electrode leads to different charging/discharging time constant throughout the electrode material. Due to the finite conductance of the electrolyte, a voltage drop is formed along the pore. The macro-pores (which are at the mouth of the pore) have much smaller time constants than the meso- and micro-pores which are located in the deeper parts of the pore. This leads to the effect that the outer parts of the pores get

charged/discharged much more quickly than the core pore structures [48, 49]. And through time, the charge stored in the macro-pores migrate to the deeper pore structure. Hence, in the case of short charging cycles, most of ions are still located at the macro-pores. The distinct open circuit voltage decay is caused by both the effect of redistribution of ions to areas of low ion concentration and self discharge, which may be caused by overpotential decomposition of the electrolyte, redox-reactions due to impurities or possible internal ohmic leakage pathway of the double-layer at the electrolyte-carbon interface.

Thus the terminal behavior of supercapacitors cannot be modeled as a single RC circuit. The charge transportation inside the porous electrodes is similar to the propagation phenomenon of infinite number of interleaved RC branches. The porous electrode theory has been used to interpret the impedance spectrum of a supercapacitor. The general equivalent circuit model of supercapacitor is composed of infinite number of parallel RC branches [50].

3.4.2 Supercapacitor VLR model

The VLR model proposed in [51] modifies the two branch model to better capture the long-term behavior of supercapacitor. It can be represented as in Figure 3.1. The first branch consists of a resistor R_1, a constant capacitor C_0 and a voltage dependent capacitor $\frac{K_V}{2} * V$. It models the instant behavior of supercapacitor, whose capacitance increases with terminal voltage. The total capacitance of the first branch is $C_1 = C_0 + \frac{K_V}{2} * V$. And the time constant is on the order of seconds. The second branch is composed of a resistor R_2 and a constant capacitor C_2. This branch models the long-term behavior of supercapacitor, especially the charge-redistribution process. The time constant of the second branch is on the order of minutes. And the third branch, which contains a voltage dependent resistor R_3, models the self-discharge process of supercapacitor.

According to the procedure described in [51], two experiments are performed to identify the VLR model parameters of a 10 F 2.7 V supercapacitor manufactured by Maxwell,

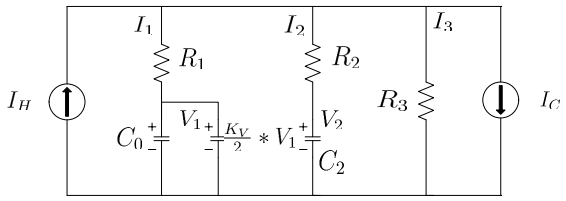


Figure 3.1: Simplified equivalent circuit model of sensor node

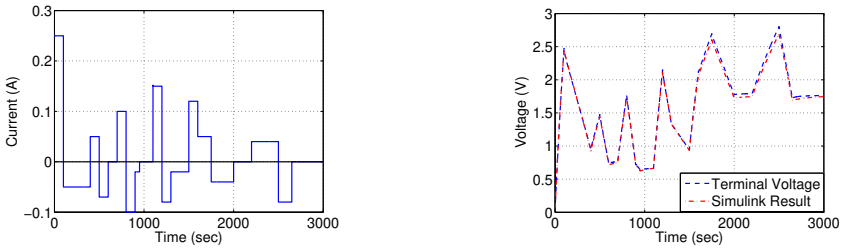
which is used as an example to validate the practical supercapacitor model presented in this paper. A Maccor testing system is used to perform these two experiments. In the charge-redistribution experiment, the supercapacitor is charged by a 1 A constant current source. Once the terminal voltage reaches the rated voltage 2.7 V, the constant current source is removed. Terminal voltage is measured during the entire process. And in the self-discharge experiment, supercapacitor is charged by a 2.7 V constant voltage source for an hour, and then the terminal voltage is recorded for 12 hours. A Maccor testing system is used to perform these two experiments. And the VLR model parameters identified with the two experiments are listed in Table 3.4. The value of variable resistor R_3 is related to terminal voltage V_{term} by piecewise linear function as follows.

$$R_3 = \begin{cases} (-5.821 \times 10^6) * V_{term} + 15.66 \times 10^6 & \text{if } V_{term} \in [0, 2.6309) \\ (-5.515 \times 10^6) * V_{term} + 14.87 \times 10^6 & \text{if } V_{term} \in [2.6309, 2.6634) \\ (-2.969 \times 10^6) * V_{term} + 8.043 \times 10^6 & \text{if } V_{term} \in [2.6634, 2.7000) \end{cases}$$

Table 3.1: Model Parameter of a 10-F Maxwell Supercapacitor

$R_1(Ohm)$	$C_0(Farad)$	K_V	$R_2(Ohm)$	$C_2(Farad)$
0.06668	7.278	2.136	139.3431	1.9137

A dynamic test is performed to validate the VLR model parameters. The supercapacitor is charged or discharged with a current profile as in Figure $3.2a$, and the terminal voltage is



(a) (b)

Figure 3.2: Dynamic test for identifing VLR model. (a) Test current profile. (b) Comparison between measured and simulated voltages.

measured by the MACCOR system. The current profile corresponds to a normal operation period of a TmoteSky MICAz sensor node [52]. When the sensor node is switched to the sleep mode, the current drawn from the supercapacitor is about 30 μA. The upper bound of discharging current is determined based on the datasheets of TmoteSky MICAz, which is set to be 100 mA. And the output current of a solar panel [53] after being conditioned by the maximum power point tracker is about 250 mA. The measured terminal voltage is then compared with the terminal voltage computed by the VLR simulink model (Figure 3.3) [51] with the same current profile input. The result in Figure 3.2b shows that the identified VLR model can capture the dynamic terminal behavior of the 10 F 2.7 V supercapacitor.

3.4.3 Supercapacitor Internal State Observer

The internal state observer of supercapacitor presented in this section uses supercapacitor terminal voltage to dynamically determine the supercapacitor internal state in a more computationally efficient way. Because the terminal voltage of supercapacitor can be easily measured on a sensor node, this approach is more practical than the VLR simulink model, which uses charging and discharging profiles as input.

Determine Supercapacitor State Using Terminal Voltage

To derive the observer of the supercapacitor internal state, we start from Figure 3.1. The energy harvesting module is modeled as a current source which outputs variable charging

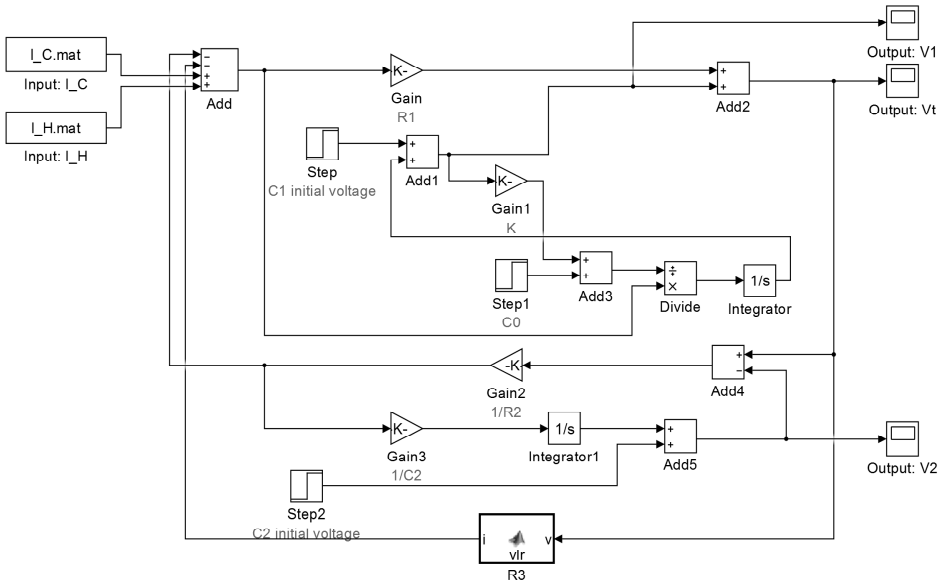


Figure 3.3: The VLR Simulink Model of Supercapacitor [51]

current I_H. The energy dissipation module is also modeled as a current source, but drains current I_C from the supercapacitor. Here we neglect the input and output regulators for simplicity's sake.

The first branch dynamics can be represented by (3.23) and (3.24),

$$I_1(t) = [C_0 + K_V V_1(t)] \frac{dV_1(t)}{dt}, \tag{3.23}$$

$$V_1(t) = V_t(t) - I_1(t) R_1. \tag{3.24}$$

From (3.23) and (3.24), we can derive an ordinary differential equation as (3.25) to represent the dynamics of the first branch.

$$\frac{dV_1(t)}{dt} = \frac{V_t(t) - V_1(t)}{R_1 C_0 + R_1 K_V V_1(t)}. \tag{3.25}$$

The numerical solution can be solved by the Euler method.

$$V_1[n] = V_1[n-1] + T * \frac{V_t[n-1] - V_1[n-1]}{R_1C_0 + R_1K_V V_1[n-1]}. \tag{3.26}$$

Thus the value of V_1 can be solved recursively using (3.26). At each time step, the current value of V_1 can be calculated given V_1 and terminal voltage at the previous time step and no complex mathematical operation is involved in the process, thus the computational cost of V_1 is mainly determined by the time step, T.

Equation (3.26) shows that the value of V_1 only depends on the history of terminal voltage. This relieves the burden of circuit design, since the magnitude of charging or discharging current is not needed for solving V_1.

Similarly, the second branch dynamics can be represented by (3.27) and (3.28),

$$I_2(t) = C_2 \frac{dV_2(t)}{dt}, \tag{3.27}$$

$$V_2(t) = V_t(t) - I_2(t)R_2. \tag{3.28}$$

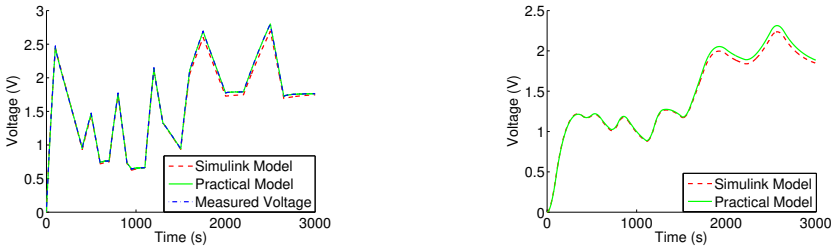
And the ordinary differential equation that describes the second branch is

$$\frac{dV_2(t)}{dt} = \frac{V_t(t) - V_2(t)}{C_2R_2} \tag{3.29}$$

V_2 can be solved using Euler method as follows.

$$V_2[n] = V_2[n-1] + \frac{T}{R_2C_2}(V_t[n-1] - V_2[n-1]). \tag{3.30}$$

At each time step, V_2 is determined by the V_2 value and terminal voltage at the previous time step, therefore computational cost is mainly determined by the length of each time interval. The value of V_2 also depends solely on the history of terminal voltage. It is



(a) V_1 (b) V_2

Figure 3.4: Dynamic test validation of the observer model

because the history of terminal voltage can reflect the change of charging or discharging current.

Since V_1 and V_2 can be calculated online using (3.26) and (3.30) given the terminal voltage at each time step, the computational cost of the proposed model is much smaller compared with the simulink model.

The proposed modeling method is validated with the dynamic charging-discharging process as shown in Figure 3.4. Assume initially the supercapacitor is fully depleted, thus the initial values of V_1 and V_2 are 0 V. The terminal voltage of the 10 F Maxwell supercapacitor characterized before is measured every 1 second and used as the input of the observer model to calculate V_1 and V_2. The step size of the observer model is 1 second.

The comparison between the observer model and the experimental measurement is shown in Figure 3.4a. It is shown that the proposed practical model can estimate V_1 with high accuracy. For comparison, V_1 calculated by the VLR simulink model under the same test condition is also plotted in Figure 3.4a. Since the supercapacitor internal state V_2 can not be directly measured, the V_2 output of VLR simulink model is used to validate that of the practical model (Figure 3.4b). This is because the difference between the simulated terminal voltage V_t and measured terminal voltage V_{term} is small in short period of time, the VLR simulink model can be used to approximate the dynamics of a supercapacitor. The results indicate the validity of the proposed practical modeling method. Figure $3.4a$ and $3.4b$ also show that the VLR simulink model outputs deviate from the experimental mea-

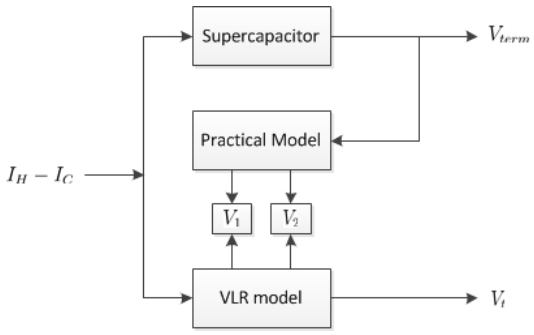


Figure 3.5: Comparison of observer model and VLR simulink model

surement and the practical model output with the increase of time. The difference between VLR simulink model and the proposed modeling method is demonstrated in Figure 3.5.

Figure 3.5 is an illustration of the VLR simulink model and the proposed practical modeling method. The input of the VLR simulink model is the charging and discharging profiles. The terminal voltage and internal states V_1 and V_2 are calculated as output. So the simulated terminal voltage V_t can be compared with the terminal voltage of supercapacitor V_{term} directly. Figure 3.2 shows that the VLR model with parameter values in Table 3.4 can capture the terminal behavior of the supercapacitor. However, the difference between V_t and V_{term} becomes larger with time, which can be seen from Figure $3.4a$. This is because the VLR model is a simplified two branch model. To capture the terminal behavior in longer period, more branches are needed.

The proposed practical method models the relationship between terminal voltage V_{term} and supercapacitor internal states V_1, V_2. During each time interval, terminal voltage is measured and then V_1 and V_2 are calculated separately. The computation of V_1 only depends on the first branch parameters and the computation of V_2 depends on the second branch parameters. Because the VLR model parameters can capture the short term behavior of the supercapacitor, practical model built based on these parameters can accurately model the dynamics within each time interval. When it comes to the next time interval, terminal voltage of the supercapacitor is measured again and used to update V_1 and V_2. Thus the error accumulation effect can be less significant compared to the VLR simulink model.

This can be shown in Figure $3.4a$. As R_1 is very small (Table 3.4), terminal voltage can be used to approximate the true value of V_1. The practical model shows a better estimation than the VLR simulink model.

Moreover, it is impractical to implement the VLR simulink model directly on a low-power wireless sensor node, while the proposed model can be easily implemented. This is because the VLR simulink model is based on coupled differential equations [54] that characterize the dynamics of the equivalent circuit model. The coupled equations create algebraic loop in the VLR simulink model, which makes it impractical on low-power embedded platforms due to high computational cost. The proposed practical model, on the other hand, is based on (3.26) and (3.30) that use only basic mathematical operations in the computation in each time slot. The model complexity is mainly determined by the step size of the practical model. By utilizing the methods presented in Section IV, the computational cost of practical model can be significantly reduced. Since it is impractical to implement the VLR Simulink model on a sensor node, the computational cost of both models is compared on a Linux machine with Intel i7-3770 3.40 GHz processor using the power measurement tool - powerstat. The average power consumption of the simulink model and the practical model is 37.4179 Watts and 35.7477 Watts respectively. The average running time is 1.1898493625 seconds for the simulink model, and 0.001307553 seconds for the practical model. Additionally, the VLR Simulink model requires tracking V_1 and V_2 values constantly, which is impossible when a sensor node is in the sleep mode. For the practical model, the initial values of V_1 and V_2 can be estimated using the method proposed in Section 3.4.3 after a sensor node wakes up.

As the proposed practical model uses numerical method, the accuracy may be a concern. Therefore, the proposed method is further investigated from two aspects to increase the accuracy with limited computing cost so that it is suitable for low-power wireless sensor nodes. One is to increase the accuracy of the numerical solution without significantly increasing the computational cost. The other is to investigate the impact of time step on the

accuracy of numerical solution. Because of the low power requirement of wireless sensor nodes, the step size of the proposed model should be set as large as possible as long as the error is acceptable.

Accuracy Improvement

In this section, predictor corrector methods [55, 56] are used to improve the accuracy of numerical solutions (3.26) and (3.30). The predictor corrector method usually contains two steps. First, the predictor provides a rough estimation of the solution using explicit method. Then in the second step, the corrector refines the estimated solution with implicit method. This method is easy to implement and can improve the accuracy of the numerical model.

To improve the accuracy of the numerical solution for $V_1(t)$ in (3.26), the Predict-Evaluate-Correct-Evaluate (PECE) method is used as follows.

$$\widetilde{V_1}[n] = V_1[n-1] + T * \frac{V_t[n-1] - V_1[n-1]}{R_1C_0 + R_1K_vV_1[n-1]} \tag{3.31}$$

$$\begin{aligned} V_1[n] = V_1[n-1] + \frac{T}{2}\{&\frac{V_t[n-1] - V_1[n-1]}{R_1C_0 + R_1K_vV_1[n-1]} \\ &+ \frac{V_t[n] - \widetilde{V_1}[n]}{R_1C_0 + R_1K_v\widetilde{V_1}[n]}\} \end{aligned} \tag{3.32}$$

Similarly, by employing PECE method, the numerical solution for $V_2(t)$ in (3.30) is calculated as

$$\widetilde{V_2}[n] = V_2[n-1] + \frac{T}{C_2R_2}(V_t[n-1] - V_2[n-1]) \tag{3.33}$$

$$V_2[n] = V_2[n-1] + \frac{T}{2}\{\frac{V_t[n-1] - V_2[n-1]}{C_2R_2} + \frac{V_t[n] - \widetilde{V_2}[n]}{C_2R_2}\} \tag{3.34}$$

The predictor corrector method has the advantage of simple implementation and low computational cost. The PECE method only adds one additional corrector step to the origi-

nal numerical solutions of V_1 and V_2, but it can generally improve the accuracy of numerical solutions as shown in Table 3.2.

Impact of Time Step

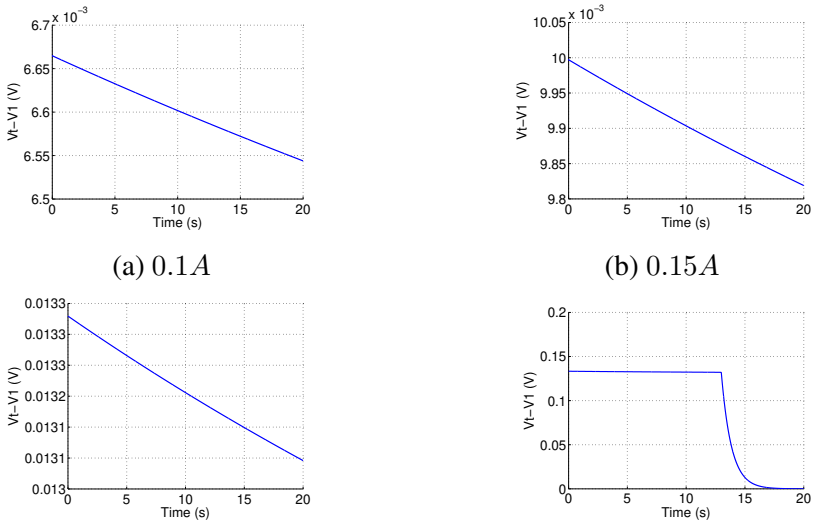
Table 3.2: Error of Practical Modeling Method with Different Time Step

Standard Error	V_1	V_2
Original (0.01)	8.5533×10^{-9}	2.5463×10^{-8}
Original + PECE (0.01)	1.1926×10^{-9}	4.4378×10^{-11}
Original (0.1)	1.4851×10^{-7}	8.0533×10^{-7}
Original + PECE (0.1)	1.7488×10^{-7}	6.0448×10^{-9}
Original (1)	1.1406×10^{-6}	2.5530×10^{-5}
Original + PECE (1)	2.0028×10^{-5}	3.5767×10^{-7}
Original (2)	0.3080	7.2371×10^{-5}
Original + PECE (2)	0.2671	1.1126×10^{-6}

Intuitively, the step size of the practical model should be set as large as possible to reduce the computational cost. In this section, the impact of the step size of practical model is explored.

The dynamic test setting in the first 1500 seconds in Figure 3.2 is employed here. The step size of simulink model is set to be 0.0001 to approximate the exact solution of V_1 , V_2 and V_t. Given the simulated terminal voltage, the practical model is investigated with step size equals to 0.01 seconds, 0.1 seconds, 1 second and 2 seconds respectively. The impact of the time step on the PECE method is also explored. The results are shown in Table 3.2.

From Table 3.2, it is shown that with the increase of step size, the accuracy of practical model decreases. When the time step of practical model equals to 2 seconds, the standard error of V_1 is no longer acceptable for practical applications. So, for a 10 F supercapacitor, an acceptable update rate for V_1 is 1 second. However, with the step size being 1 second, the standard error of V_2 is still very small. Thus we investigate computing V_1 and V_2 separately in the next section.



(a) $0.1A$ (b) $0.15A$

(c) $0.2A$ (d) $2A$

Figure 3.6: Deviation between terminal voltage and V1 when charging current equals to (a) $0.1A$, (b) $0.15A$, (c) $0.2A$, and (d) $2A$.

Switching Policy for V_1

Given that R_1 is very small, it seems reasonable to assume V_1 equals to terminal voltage to avoid the calculation of V_1. But this contradicts with the assumption that during rapid charging or discharging process, current is mainly injected into or drawn from the first branch, which means I_1 is either greater than zero or less than zero. But $V_1 = V_t$ implies $I_1 = 0$. Therefore, it is necessary to analyze under what situation V_1 can be approximated by the terminal voltage.

First of all, it is obvious that $V_1 \neq V_t$ when there is large charging or discharging current, because the voltage drop on R_1 becomes significant. We define δ_I as the threshold for the magnitude of charging or discharging current. When $|I_H - I_C| > \delta_I$, the terminal voltage cannot be used to approximate V_1, and V_1 has to be calculated numerically with a time step of at least 1 second according to Table 3.2. We conducted a series of simulations to investigate the impact of charging current on the deviation between terminal voltage and V_1. The results are shown in Figure 3.6. The impact of discharging current shows similar results.

Secondly, consider the case when the supercapacitor is charged by a large constant current source. After the first branch is fully charged, less current is injected to the first branch and more current flows to the second branch. With the magnitude of I_1 decreasing, voltage drop on R_1 become less significant. Moreover, the supercapacitor is often equipped with charge protection mechanism in practice, which disconnects the power source once supercapacitor terminal voltage reaches its rated voltage.

Thus charging or discharging time is also a factor that influence the difference between V_1 and terminal voltage. The time threshold δ_t can be calculated as the time for the first branch to be fully charged or fully discharged. During the charging process, if the charging time exceeds δ_t, then the first branch is fully charged and the deviation of V_1 from terminal voltage becomes less significant and we can use terminal voltage to approximate V_1. For the discharging process, if the discharging time exceeds δ_t, the first branch is fully depleted and the difference between V_1 and terminal voltage becomes negligible, therefore terminal voltage can be used to estimate V_1. For the case with 2 A charging current (Figure $3.6d$), δ_t can be calculated as

$$\begin{aligned}\delta_t &= \frac{(C_0 + \frac{K_v}{2} * 2.7) * 2.7}{I_H - I_C} \\ &= \frac{(7.278 + \frac{2.136}{2} * 2.7) * 2.7}{2} \\ &= 13.7182.\end{aligned} \tag{3.35}$$

When $t > 13.72$, the difference between V_1 and V_t becomes negligible because the first branch has been fully charged.

From the above investigation, we can set $\delta_I = 150$ mA, if the error tolerance is 0.01 V. And δ_t is set to be the time when the first branch capacitor is fully charged or depleted, if the charge/discharge current exceeds δ_I.

The algorithm formulation of the switching policy is shown in Algorithm 4. A dynamic test is used to evaluate the validity of the proposed method. The charging and discharging

profile is the same as the first 1500 seconds in Figure 3.2. The result (Figure 3.7) indicates that the proposed method can capture the dynamics of V_1 without computing V_1 every 1 second.

Algorithm 1 Switching Policy for Computing V_1

Require: charging current $I_H - I_C$ and terminal voltage V_t

$\delta_t = 0$;
$\delta_I = 0.15$;
$t = 0$;
$flag = 1$;
1: **if** $|I_H - I_C| > \delta_I$ **then**
2: $t = t + T$;
3: **if** $flag = 1$ **then**
4: $V_1 = V_1[n-1]$;
5: $flag = 0$;
6: **if** $I_H - I_C > 0$ **then**
7: $\delta_t = \frac{(C_0+\frac{K_v}{2}*2.7)*2.7-(C_0+\frac{K_v}{2}*V_1)*V_1}{I_H-I_C}$;
8: **else**
9: $\delta_t = \frac{(C_0+\frac{K_v}{2}*V_1)*V_1}{|I_H-I_C|}$;
10: **end if**
11: **end if**
12: **if** $t < \delta_t$ **then**
13: V_1[n]=NumericalEquation(V_1[n-1]);
14: **else**
15: $V_1 = V_t$;
16: $flag = 1$;
17: **end if**
18: **else**
19: $V_1 = V_t$;
20: $t = 0$;
21: $flag = 1$;
22: **end if**

Exploring Step Size for V_2

As V_1 and V_2 are calculated separately and the update rate of V_1 can be reduced with switching policy, we further increase the step size for calculating V_2 in this section.

When the time step is further increased, the increase of V_2 standard error is illustrated in Table 3.3. It is shown in Figure $3.8a$ that when the time step is increased to as large as 50

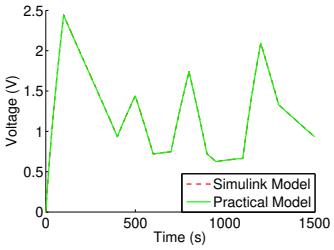
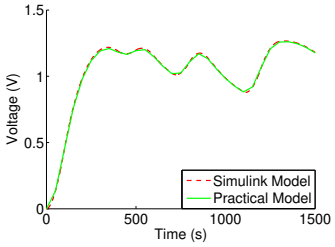


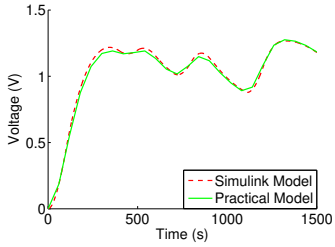
Figure 3.7: Comparison between V_1 predicted by the practical model with switching policy and that by the VLR simulink model.

Table 3.3: Standard error of numerical method with larger time intervals

Step Size	Original	Original+PECE
10 seconds	8.2505×10^{-4}	2.1330×10^{-5}
20 seconds	0.0024	9.3234×10^{-5}
30 seconds	0.0045	5.9889×10^{-4}
40 seconds	0.0072	0.0015
50 seconds	0.0106	8.2143×10^{-4}
60 seconds	0.0142	0.0033
70 seconds	0.0186	0.0048



(a) 50 seconds



(b) 60 seconds

Figure 3.8: Comparison between V_2 predicted by the simulink model and that by the practical model with step size equals to (a) 50 seconds, and (b) 60 seconds.

seconds, with PECE method the practical model can still capture the dynamics of V_2. This is because the second branch has a much larger time constant, which is about 4 minutes. So the change of V_2 can still be captured even when the numerical solution is updated every 50 seconds. But when the time step is increased to 60 seconds, the error of the practical model becomes significantly large (Figure $3.8b$).

Initial Value Estimation

One important issue with the proposed practical model is determining the initial values of V_1 and V_2. Because V_1 and V_2 are not directly measurable, and it is impossible to keep track of the supercapacitor state all the time in a real sensor node, especially during the sleep mode, an initial value estimation method is proposed to calculate V_1 and V_2 when the system is awaken.

After waking up a sensor node, the initial value of V_1 can be approximated by the terminal voltage of supercapacitor. Given this approximation and the measured terminal voltage, V_1 value at the next time interval can be calculated using (3.26). By iteratively executing (3.26) using the calculated V_1 value and measured terminal voltage at the previous time interval, V_1 value can be updated sequentially for each time interval, and the error caused by initial value approximation can be reduced. Usually after 10 updates, the accuracy of V_1 is sufficient, which can then be used to estimate the initial value of V_2. If the step size for updating V_1 is 1 second, then after 10 seconds V_1 calculated from the practical model can be used to approximate the exact solution of V_1 and calculate the initial value of V_2. The time interval for updating V_1 before calculating the initial value of V_2 is defined as Δt for general purpose.

Once the value of V_1 is determined, it is used to estimate the initial value of V_2 through

$$V_2[n] = (1 + \frac{R_2}{R_1} + \frac{R_2}{R_3})V_t[n] - (I_H[n] + I_C[n])R_2 - \frac{R_2}{R_1}V_1[n]. \tag{3.36}$$

(3.55) is derived from the following equations:

$$I_2 = I_H - I_C - I_1 - I_3 = I_H - I_C - \frac{V_t - V_1}{R_1} - \frac{V_t}{R_3}, \tag{3.37}$$

$$V_2 = V_t - I_2 R_2. \tag{3.38}$$

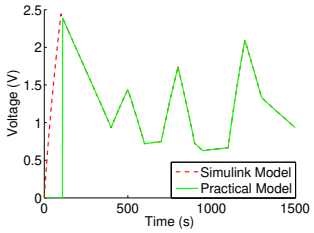
Because $\frac{R_2}{R_1}$ generally has large value, the error of V_1 estimation can be magnified. That is why (3.55) can not be directly used to compute the value of V_2 all the time. After the initial value of V_2 is computed, the practical modeling method for V_2 in (3.30) can further reduce the estimation error caused by initial value approximation.

A dynamic test with the same setting as the first 1500 seconds of the current profile in Figure 3.2 is used to validate our proposed initial value estimation method. Δt is set to be 10 seconds. Initially V_1 and V_2 are set to be zero. When $t = t_1$, V_1 is approximated by the terminal voltage and then the practical modeling method is employed to compute V_1. After 10 seconds ($t_2 = t_1 + 10$), the initial value of V_2 is calculated using (3.55), and then the practical modeling method is used to calculate V_2 based on the estimated initial value. The step size of the practical model is set to be 50 seconds for V_2.

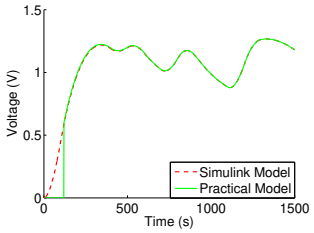
The simulation results are shown in Figure 3.9 and Figure 3.10. In Figure 3.9, the estimation of V_1 starts when $t = t_1 = 110$ s, then after 10 s when $t = t_2 = 120$ s, the estimation of V_2 starts. In the simulation shown in Figure 3.10, the time when V_1 estimation starts is $t_1 = 610$ s and the time when V_2 estimation starts is $t_2 = 620$ s. The results indicate that the initial values of V1 and V2 can be accurately determined using the proposed method.

3.4.4 Available Charge and Energy Estimation Based on Practical Model

Based on the proposed practical model, internal state of supercapacitor can be determined. Thus the quickly and not quickly available charge and energy that are stored in the super-

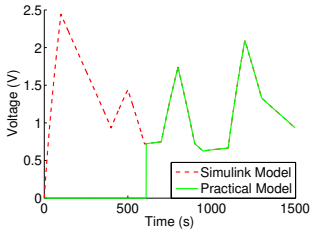


(a) V_1 estimation started at 110 s

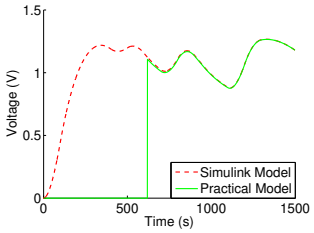


(b) V_2 estimation started at 120 s

Figure 3.9: Initial value estimation evaluation result 1.



(a) V_1 estimation started at 610 s



(b) V_2 estimation started at 620 s

Figure 3.10: Initial value estimation evaluation result 2.

capacitor can be estimated.

The charge stored in the first branch capacitor is

$$Q_{QA} = [C_0 + \frac{K_V}{2} V_1(t)]V_1(t). \tag{3.39}$$

This part of charge is quickly available (QA).

The charge stored in the second branch capacitor is

$$Q_{nQA} = C_2 V_2(t), \tag{3.40}$$

which is not quickly available (nQA).

When $V_2 - V_1 > 0$, charge stored in the second branch will transfer to the first branch. When $V_1 - V_2 > 0$, charge stored in the first branch will transfer to the second branch. The end of redistribution process corresponds to the state $V_1 = V_2$. And the value of V_1 and V_2 can be calculated as follows.

Assume the total charge at the beginning of redistribution ($time = 0$) equals to the total charge at the end of redistribution ($time = t$). The charge dissipated through the third branch can be neglected considering R_3 is very large.

$$[C_0 + \frac{K_V}{2}V_1(0)]V_1(0) + C_2V_2(0) = [C_0 + \frac{K_V}{2}V_1(t)]V_1(t) + C_2V_2(t). \tag{3.41}$$

Let $V_1(t) = V_2(t) = X$, then we have

$$\frac{K_V}{2}X^2 + (C_0 + C_2)X - [C_0 + \frac{K_V}{2}V_1(0)]V_1(0) - C_2V_2(0) = 0. \tag{3.42}$$

By solving the above equation, the values of V_1 and V_2 at the end of redistribution can be solved.

During this process, the transferred charge is

$$Q_t = \frac{1}{R_1 + R_2}\int_0^t (V_2(s) - V_1(s))ds, \tag{3.43}$$

V_1 and V_2 can be calculated from (3.25) and (3.29):

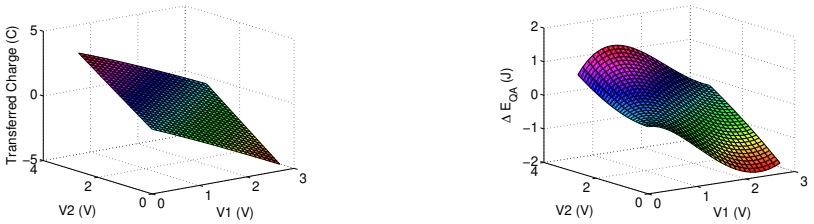
$$V_2 = V_t - C_2R_2\frac{dV_2}{dt}, \tag{3.44}$$

$$V_1 = V_t - (R_1C_0 + R_1K_vV_1)\frac{dV_1}{dt}. \tag{3.45}$$

Plugging (3.44) and (3.45) into (3.43),

$$\begin{aligned} Q_t = -\frac{C_2R_2}{R_1 + R_2}[V_2(t) - V_2(0)] + \frac{R_1C_0}{R_1 + R_2}[V_1(t) - V_1(0)] \\ + \frac{R_1K_V}{2(R_1 + R_2)}[V_1^2(t) - V_1^2(0)]. \end{aligned} \tag{3.46}$$

Based on (3.46), the transferred charge under all the possible initial states of superca-



(a) Charge Transfer During Charge Redistribution

(b) Energy Change During Charge Redistribution

Figure 3.11: Significance of Charge Redistribution.

pacitor are shown in Figure $3.11a$.

From Figure $3.11a$, we can see charge redistribution phenomenon has significant impact on power management of sensor nodes. Considering the case when $V_1 = 1.2$ V, $V_2 = 2$ V, at the end of redistribution, 1.285 C charge will be transferred to the first branch and become quickly available. This amount of charge can be used to execute tasks for 12.85 seconds if the average current drawn is 100 mA.

The energy stored in the first branch can be calculated as

$$E_{QA}(t) = \int_0^t (C_0 + K_V V_1) V_1 dV_1 = \frac{1}{2} C_0 V_1^2(t) + \frac{1}{3} K_V V_1^3(t). \tag{3.47}$$

This part of energy can be quickly retrieved from the supercapacitor.

The energy stored in the second branch is

$$E_{nQA}(t) = \frac{1}{2} C_2 V_2^2(t). \tag{3.48}$$

And it takes some time for the second branch energy to be recovered.

Therefore, the remaining energy in a supercapacitor can be estimated based on its state. Moreover, the upper limit of energy change caused by charge redistribution can be estimated.

The change of E_{QA} can be calculated as

$$\Delta E_{QA}(t) = \frac{1}{2}C_0V_1^2(t) + \frac{1}{3}K_VV_1^3(t) - \frac{1}{2}C_0V_1^2(0) - \frac{1}{3}K_VV_1^3(0), \tag{3.49}$$

in which, $V_1(t)$ can be calculated from (3.41).

Figure $3.11b$ shows the upper bound of energy gain during the charge redistribution process. It is shown that when $V_1 = 0.7$ V and $V_2 = 2.7$ V, the change of E_{QA} reaches maximum, which is 1.629 J. When $V_1 = 1.2$ V and $V_2 = 2$ V, the upper bound of the energy obtained from charge redistribution is 0.7447 J. If $V_1 = 2$ V and $V_2 = 1.2$ V, at most 0.6972 J of energy will become not quickly available after charge redistribution.

Since the practical model can be used to estimate the current quickly available charge and energy, and further predict how much charge and energy can be obtained by taking advantage of charge redistribution phenomenon, it can provide an important guide for developing energy aware power management techniques and application rate control algorithms for supercapacitor powered sensor nodes.

3.4.5 Supercapacitor Power Input Model

In many applications, the supercapacitor is connected to the system via an interface circuit, which is characterized by a variable energy conversion efficiency. Thus, a model that describes the dynamic behavior of supercapacitor fed by a dynamic charging/discharging power profile is important for developing power management techniques.

Dynamic power-fed supercapacitor behavior

To derive the supercapacitor model that takes power as input, the VLR equivalent circuit model is employed. According to Kirchhoff's current law, the relationship of the three branch currents can be described by

$$I_1 + I_2 + I_3 = I_H - I_C. \tag{3.50}$$

The relationship between supercapacitor terminal voltage V_t and its internal state V_1 and V_2 can be derived as

$$\begin{aligned} V_t &= V_1 + I_1 R_1 = V_1 + (-I_2 - I_3 + I_H - I_C) R_1 \\ &= V_1 + (-\frac{V_t}{R_3} - \frac{V_t - V_2}{R_2}) R_1 + (I_H - I_C) R_1 \\ &= V_1 - \frac{R_1}{R_3} V_t - \frac{R_1}{R_2}(V_t - V_2) + (I_H - I_C) R_1 \\ &= V_1 + \frac{R_1}{R_2} V_2 - (\frac{R_1}{R_3} + \frac{R_1}{R_2}) V_t + (I_H - I_C) R_1. \end{aligned} \tag{3.51}$$

V_t can be calculated by

$$V_t = R_M [V_1 + \frac{R_1}{R_2} V_2 + (I_H - I_C) R_1], \tag{3.52}$$

where

$$R_M = \frac{R_2 R_3}{R_2 R_3 + R_1 R_2 + R_1 R_3}. \tag{3.53}$$

Since charging/discharging current can be represented by

$$I_H(t) - I_C(t) = \frac{P(t)}{V_t(t)}, \tag{3.54}$$

(3.52) can be translated into

$$V_t^2(t) - R_M (V_1(t) + \frac{R_1}{R_2} V_2(t)) V_t(t) - R_M R_1 P(t) = 0. \tag{3.55}$$

The solution of (3.55) is

$$V_{est}(t) = \frac{R_M}{2}(V_1(t) + \frac{R_1}{R_2} V_2(t)) + \sqrt{\frac{1}{4} R_M^2 (V_1(t) + \frac{R_1}{R_2} V_2(t))^2 + R_M R_1 P(t)}. \tag{3.56}$$

Here, $V_t(t)$ is replaced by $V_{est}(t)$, which is used to represent the estimated terminal voltage value.

Based on (3.26) and (3.30), the supercapacitor model that takes charging/discharging power as input can be derived by incorporating

$$V_{est}[n] = \frac{R_M}{2}(V_1[n] + \frac{R_1}{R_2}V_2[n]) + \sqrt{\frac{1}{4}R_M^2(V_1[n] + \frac{R_1}{R_2}V_2[n])^2 + R_M R_1 P[n]}. \quad (3.57)$$

In which, $R_M = \frac{R_2 R_3}{R_2 R_3 + R_1 R_2 + R_1 R_3}$ and P denotes the charging/discharging power. At each time instant, the value of V_1 and V_2 are first estimated based on their values at previous instant and then used to estimate the value of V_{est}. Iterate (3.26), (3.30) and (3.57) with V_{est} replacing V_t predicts the terminal voltage based on the charging/discharging power at each time instant.

Thus, assuming the supercapacitor state (V_1 and V_2) at any time t is given, the terminal voltage of the supercapacitor V_{est} at time t can be estimated. Furthermore, the supercapacitor state at time $t+\Delta t$ can be calculated from $V_{est}(t)$ based on (3.26) and (3.30). Therefore, terminal voltage of supercapacitor can be recursively predicted as long as the power input $P[n]$ is given for each time step.

At each time slot, terminal voltage is predicted using simple mathematical operations. Therefore, the computational cost of the proposed modeling method is mainly determined by the step size, T.

What is to be noticed is that the value of R_M has to be determined before calculating the terminal voltage V_{est} as in (3.56). While, to calculate R_M, the value of R_3 is needed, which is a piecewise linear function of terminal voltage V_{est}. To solve this problem, the estimated terminal voltage V_{est} in the previous time slot is used to determine R_3, and then calculate for R_M. This approximation is reasonable if the step size T is not very large. As a matter of fact, the variance of R_3 has been shown to have little impact on the short-term and mid-term behavior of supercapacitor.

The model proposed in our previous work [57] estimates the supercapacitor state of charge given its terminal voltage measurement. The model developed in this work intends

to use the predicted charging power profile (for energy harvesting sensor nodes) and estimated power consumption of various tasks as input and predict the terminal voltage of supercapacitor. The purpose of the power input model is to support many power management applications in which the terminal behavior needs to be predicted based on the desired working status of sensor node and the estimated harvesting energy. These power management algorithms often use the predicted terminal voltage to decide if the power consumption of the sensor node needs to be adjusted to maintain energy neutral operation, since the supercapacitor terminal voltage needs to be larger than a threshold to guarantee minimum operation of the sensor node.

Moreover, the model in the previous section can be used in conjunction with the model developed in this section to conduct receding horizon power management [13]. At each planning interval, the current state of charge could be estimated with the measured terminal voltage, and then used as the initial condition for predicting the terminal behavior for a finite horizon. Optimal power management strategy can be calculated for the finite horizon. Then at the next time interval, this process can be repeated to calculate the optimal strategy for the next finite horizon. With the receding horizon power management, the influence of model uncertainty and perturbation can be greatly reduced. Furthermore, the developed power input model can also be used in the simulations to study the characteristics, especially the charge redistribution, of supercapacitor fed by a dynamic charging/discharging power profile.

Model parameter identification of supercapacitor

A parameter identification method is proposed for the VLR model in the work of Yang[54]. The parameter identification procedures are based on two experiments: a charge-redistribution experiment for the first and second branches, and a self-discharge experiment for the leakage branch. The identified model parameters will then be further validated by a dynamic test, which includes several charging and discharging processes. The Maccor system [58]

is used to perform the three experiments.

However, the model parameter identification method in previous work is based on three assumptions. The first assumption is that there is no interaction among branches during rapid charging or discharging, which is generally not true. The second one assumes there is no leakage during the charge redistribution experiment. But self-discharge is constantly happening as long as the terminal voltage of supercapacitor is not zero. The third assumption assumes the charge redistribution stops at $t = 3\tau_2$, in which τ_2 is the time constant of the second branch. This indicates the voltages of first branch and second branch achieve balance at time $3\tau_2$. It often requires many trial and error to find an appropriate value for $3\tau_2$. Therefore, multiple trials have to be conducted to find a good set of parameters.

To bypass the assumptions of previous work and facilitate the identification process, a genetic algorithm based identification method is presented. With the proposed method, a dynamic test is performed to identify the first and second branch parameter values and a self-discharge experiment is performed to identify the variable leakage resistance. Then a different dynamic test has to be conducted to validate the identified parameters. For differentiation purpose, the first dynamic test profile is called the training profile and the second dynamic test profile is the testing profile.

The genetic algorithm is used to find parameter values that minimizes the fitness function that will be defined later. To reduce the search space of the genetic algorithm, the following constraints are added based on the supercapacitor characteristics.

1. R_1 can be estimated given the ESR value from the data sheet. The constraint for R_1 is

$$R_1 \in [ESR - \delta_1, ESR + \delta_1], \tag{3.58}$$

in which, δ_1 can be chosen as $\delta_1 = 0.02 * ESR$. This is based on the fact that R_1 is very similar to the value of ESR.

2. The constraint for the first branch capacitance is

$$C_0 < C_{rated} \tag{3.59}$$

and

$$C_0 + \frac{K_v}{2} \times V_{rated} + C_2 > C_{rated}. \tag{3.60}$$

This is because C_0 corresponds to the minimum capacitance and $C_0 + K_v \times V_{rated} + C2$ corresponds to the maximum capacitance of supercapacitor. The capacitance range of supercapacitor must contains the rated capacitance.

3. The constraint for second branch capacitance is

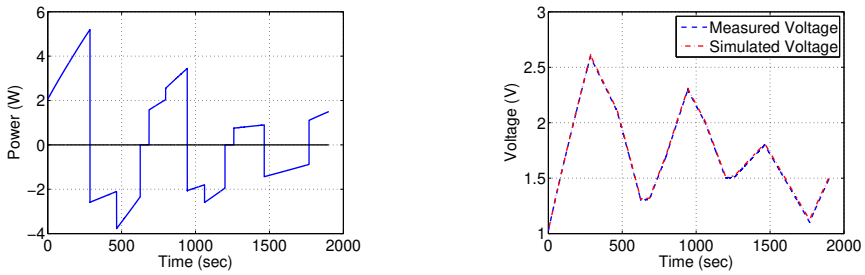
$$C_2 < C_0. \tag{3.61}$$

This constraint is due to the fact that the redistributed charge is only a small portion of the total charge. Because of the finite conductance of electrolyte and the small size of micro and meso pores, charge stored in these pores are limited. Therefore, C_2, which represents the storing capacity of these pores, must be smaller than C_0.

4. The time constants of first and second branches must satisfy

$$\frac{(C_0 + K_v \times V_1)R_1}{C_2 R_2} < 0.01. \tag{3.62}$$

This constraint is used to make sure the ratio of first and second branch time constant is smaller than 0.01, thus the two branches represent the immediate dynamic and delayed dynamic of supercapacitor respectively.



(a) (b)

Figure 3.12: Train the power input model using dynamic test. (a) Training power profile. (b) Comparison between measured and simulated voltages.

The optimization problem can be formulated as

$$\begin{aligned} \underset{X}{\text{minimize}} \quad & \text{fitness}(X) \\ \text{subject to} \quad & C_0 < C_{rated}, \\ & C_0 + \frac{K_v}{2} \times V_{rated} + C_2 > C_{rated}, \\ & C_2 < C_0, \\ & \frac{(C_0 + K_v * V_1) * R_1}{C_2 R_2} < 0.01. \end{aligned} \tag{3.63}$$

In (3.63), X is a vector that represents the set of parameters $(R_1, R_2, C_0, K_v, C_2)$. $\text{fitness}(X)$ represents the squared error between the measured voltage and the voltage estimated by the model using the parameter set X.

A 310 F 2.7 V supercapacitor is used to validate both the parameter identification method and the supercapacitor power input model. The dynamic charging/discharging power profile used for training is shown in Figure 3.12a. Before the experiment, the supercapacitor is charged by a 1 volt constant voltage source for an hour, therefore the initial values of V_1 and V_2 are 1 V. The terminal voltage of supercapacitor during the test is measured and shown in Figure 3.12b.

Since the genetic algorithm is a random search algorithm in nature, we run the genetic algorithm for 25 times. For each run, the population of each generation is 500. The maxi-

mum number of generations can be 200. But the algorithm will be terminated if the fitness value no longer shows improvement. The set of parameters corresponding to the smallest fitness value is selected as the parameter for the 310 F 2.7 V supercapacitor, which is shown in Table 3.4. The voltage simulated by the power input model is compared with the measured data as in Figure 3.12b, which indicates that the identified model parameters can represent the terminal behavior of the supercapacitor.

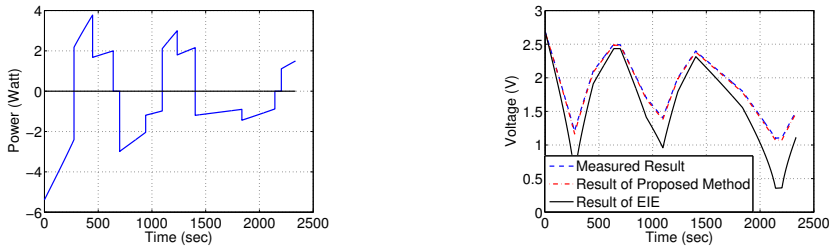
Table 3.4: Model Parameter of a 310 F 2.7 V Supercapacitor

$R_1(Ohm)$	$R_2(Ohm)$	$C_0(Farad)$	K_V	$C_2(Farad)$
0.00224	10	298.37960	29.994	12.077

After determining the parameters of first and second branches, the value of variable resistor R_3 is related to the terminal voltage V_{term} by a piecewise linear function as in (3.64).

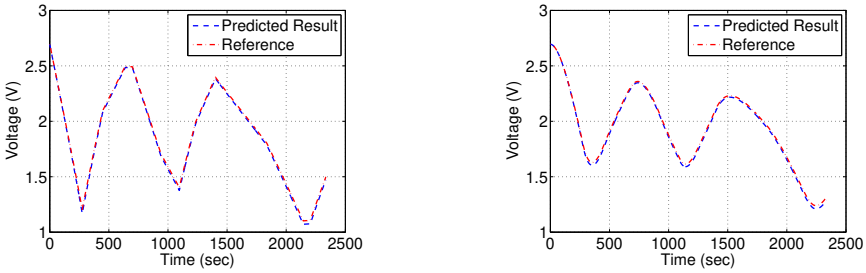
$$R_3 = \begin{cases} (-3190) * V_{term} + 8831 & \text{if } V_{term} \in [2.628, 2.7) \\ (-6342) * V_{term} + 1.711 \times 10^4 & \text{if } V_{term} \in [2.574, 2.628) \\ (-1.044 \times 10^4) * V_{term} + 2.766 \times 10^4 & \text{if } V_{term} \in [2.552, 2.574) \\ (-1.683 \times 10^4) * V_{term} + 4.387 \times 10^4 & \text{if } V_{term} \in [2.488, 2.552) \\ (-4.773 \times 10^4) * V_{term} + 1.202 \times 10^5 & \text{if } V_{term} \in [2.379, 2.488) \\ (-2.082 \times 10^5) * V_{term} + 5.009 \times 10^5 & \text{if } V_{term} \in [0, 2.379) \end{cases} \tag{3.64}$$

The identified model parameters are validated using a testing power profile as in Figure 3.13a. The initial state of the supercapacitor is $V_t = V_1 = V_2 = 2.7V$. This is achieved by charging the supercapacitor with a 2.7 V constant voltage source for one hour before the dynamic test. The comparison of the simulated voltage and the measured voltage is shown in Figure 3.13b. It is shown that the terminal behavior of supercapacitor can be accurately predicted by the developed power input model with the identified parameters. For comparison, the supercapacitor terminal behavior predicted by the energy iteration equation (EIE) model is also shown in Figure 3.13b. EIE model significantly underestimate



(a) (b)

Figure 3.13: Dynamic test validation of the power input model. (a) Testing power profile. (b) Comparison between measured and simulated voltages.



(a) V_1 comparison. (b) V_2 comparison.

Figure 3.14: Comparison of the Supercapacitor internal state

the terminal voltage since the charge redistribution phenomenon is neglected. Moreover, the internal state of supercapacitor is estimated from the measured terminal voltage using the numerical method proposed in previous literature [57]. The internal state variables (V_1 and V_2) are then compared with that estimated from the method in Section 3.4.5. The results are shown in Figure 3.14. It indicate that the internal states can also be captured by the proposed modeling method. In the following sections, this model will be used to analyze the influence of charge redistribution on power management.

Charge Redistribution Analysis of Supercapacitors with Different Rated Capacitance

Based on the developed model, two figures of merit can be defined for evaluating charge redistribution in supercapacitor. To derive the figures of merit, we first calculate the currents that flow into the two branches, since the currents of immediate branch and delayed branch represent the speed of charge accumulation in the two branches.

According to (3.26) and (3.30), the relation between the first branch current I_1 and the second branch current I_2 can be derived as:

$$I_1 = \frac{V_t - V_1}{R_1} = \frac{I_2 R_2 + V_2 - V_1}{R_1} = I_2 \frac{R_2}{R_1} - \frac{V_1 - V_2}{R_1}. \tag{3.65}$$

Plug (3.65) into (3.50) gives:

$$(\frac{R_2}{R_1} + 1) I_2 - \frac{V_1 - V_2}{R_1} = I_H - I_C - I_3. \tag{3.66}$$

Thus, I_2 can be solved as

$$I_2 = \frac{R_1}{R_1 + R_2}(I_H - I_C - I_3) + \frac{V_1 - V_2}{R_1 + R_2}, \tag{3.67}$$

and I_1 can be solved using the similar way as

$$I_1 = \frac{R_2}{R_1 + R_2}(I_H - I_C - I_3) - \frac{V_1 - V_2}{R_1 + R_2}. \tag{3.68}$$

Equation (3.26) and (3.30) show that the currents injected into the first and second branch are composed of two components. The first one is the result of charging/discharging current minus the self-discharge current. The second one is caused by the charge redistribution effect. For example, on the right hand side of (3.67), the first term is the result of external charging/discharging, and the second term is the result of charge redistribution. When $V_1 > V_2$, the charge transfers from first branch to second branch, which makes the second term positive. While when $V_1 < V_2$, charge transfers from second branch back to first branch. And the second term becomes negative.

Let

$$K_c = \frac{R_2}{R_1 + R_2}, \tag{3.69}$$

then

$$1 - K_c = \frac{R_1}{R_1 + R_2}. \tag{3.70}$$

K_c represents the portion of charge injected into the first branch. Larger K_c means more charge flow into the first branch and less charge flow into the second branch. This causes the unbalance of charge storage in the first two branches.

From the second term on the right hand side of (3.67) and (3.68), the second charge redistribution related figure of merit can be derived. It reflects the charge transfer between the first two branches. As is shown, the redistribution current depends on $|V_1 - V_2|$. Moreover, the the redistribution current is monotonically decreasing, since the value of $|V_1 - V_2|$ is decreasing. Therefore, the decreasing rate of $|V_1 - V_2|$ can reflect the significance of charge redistribution.

The dynamic change of $V_1 - V_2$ is analyzed as follows with two assumptions. First, assume there is no external charging or discharging to focus on redistribution between the first two branches. Second, the effect of self-discharge is neglected because of the small magnitude of self-discharge current. With these assumptions, first branch dynamic can be represented by

$$\frac{V_2 - V_1}{R_1 + R_2} = (C_0 + K_v V_1)\frac{dV_1}{dt}, \tag{3.71}$$

and the second branch dynamic can be described as

$$\frac{V_1 - V_2}{R_1 + R_2} = C_2 \frac{dV_2}{dt}. \tag{3.72}$$

Thus the dynamic of $V_1 - V_2$ can be derived as

$$\begin{aligned} \frac{d(V_1 - V_2)}{dt} &= \frac{dV_1}{dt} - \frac{dV_2}{dt} \\ &= \frac{V_2 - V_1}{(R_1 + R_2)(C_0 + K_v V_1)} + \frac{V_2 - V_1}{(R_1 + R_2)C_2} \\ &= \frac{V_2 - V_1}{R_1 + R_2}\left(\frac{1}{C_0 + K_v V_1} + \frac{1}{C_2}\right). \end{aligned} \tag{3.73}$$

Let

$$K_r = \frac{1}{R_1 + R_2}(\frac{1}{C_0 + K_v V_1} + \frac{1}{C_2}), \tag{3.74}$$

then (3.73) can be converted to

$$\frac{d(V_1 - V_2)}{dt} = -K_r(V_1 - V_2). \tag{3.75}$$

If K_r was a constant, the solution of $V_1(t) - V_2(t)$ is

$$V_1(t) - V_2(t) = [V_1(0) - V_2(0)]e^{-K_r t}, \tag{3.76}$$

where K_r represents the rate of decay of $V_1(t) - V_2(t)$. Here, K_r is a function of V_1, which is not a constant. The value of V_1 changes within $[1.0, 2.7]$, since in practice a DC-DC converter is often used to stabilize the output voltage of the supercapacitor, which has a minimum input voltage of 1 volt. Although K_r is not a constant, it still represents the rate of balancing of two branch capacitors, which also reflects the significance of charge redistribution.

In summary, the significance of charge redistribution can be described by K_c and K_r. K_c represents the portion of current injected into or extracted from the first and second branches. It reflects the significance of charge redistribution from the charge accumulation or extraction point of view. K_r represents the rate of balancing of $V_1(t)$ and $V_2(t)$, which reflects the significance of charge redistribution from the internal charge balancing point of view. In the next section, it will be shown that for supercapacitors with different rated capacitance, K_c and K_r have similar values. Thus different types of supercapacitors share similar charge redistribution phenomenon.

In this section, the charge redistribution phenomenon of supercapacitors with various rated capacitance are studied based on the proposed figures of merit. The investigated supercapacitors are manufactured by Maxwell with a rated capacitance of 5 F, 10 F, 50 F,

100 F, 150 F and 310 F.

The parameter identification method proposed in Section 3.4.5 is employed to identify the parameters for all the supercapacitors. Since the parameters can be different even for supercapacitors with the same rated capacitance due to the manufacture process, two samples are tested for each type of supercapacitor (three samples for 5 F supercapacitor), and the average value of the parameters are calculated to reveal the common feature for the type of supercapacitor. Using the average values, K_c and K_r are calculated to evaluate the significance of charge redistribution for the corresponding type of supercapacitor. What is to be noticed is that, K_r depends on the value of V_1. Thus three K_r are calculated. Max K_r represents the value corresponding to $V_1 = 1$ V. Min K_r corresponds to $V_1 = 2.7$ V. Avg K_r is calculated when $V_1 = 1.85$ V, which is the median of V_1's working region. The results are shown in Table 3.5.

Table 3.5: Charge Redistribution Analysis of Different Type of Supercapacitor

Supercapacitor Model	R_1	R_2	C_0	K_v	C_2	K_c	Max K_r	Avg K_r	Min K_r
5 F 2.7 V 1	0.17307	393	3.48084	1.18291	1.23364				
5 F 2.7 V 2	0.17193	229	3.97187	1.00311	1.66992				
5 F 2.7 V 3	0.17205	109	4.86758	0.56987	2.51478				
Average	0.17235	243.66667	4.10676	0.91863	1.80611	0.99929	0.00309	0.00298	0.00289
10 F 2.7 V 1	0.07593	73	7.64376	1.51538	2.16922				
10 F 2.7 V 2	0.07374	68	5.80484	2.16775	1.26870				
Average	0.07488	69.5	6.58441	1.90187	1.74104	0.99894	0.00990	0.00964	0.00945
50 F 2.7 V 1	0.02029	20	46.11091	4.84665	17.34194				
50 F 2.7 V 2	0.02040	54	47.98547	3.60377	11.43605				
Average	0.02034	37	47.04819	4.22521	14.38900	0.99945	0.00240	0.00237	0.00234
100 F 2.7 V 1	0.01484	14	82.29080	15.56573	22.99865				
100 F 2.7 V 2	0.01494	42	73.40343	19.14485	4.45104				
Average	0.01489	28	77.84712	17.35529	13.72485	0.99947	0.00298	0.00293	0.00289
150 F 2.7 V 1	0.01411	19	98.29125	29.64278	19.71274				
150 F 2.7 V 2	0.01423	19	101.09580	27.62440	13.20188				
Average	0.01417	19	99.69353	28.63359	16.45731	0.99925	0.00361	0.00354	0.00349
310 F 2.7 V 1	0.00224	10	298.37960	29.99440	12.07665				
310 F 2.7 V 2	0.00247	7	309.96540	29.96536	99.78468				
Average	0.00236	8.5	304.17250	29.97988	55.93067	0.99972	0.00245	0.00243	0.00241

From Table 3.5, it is obvious that even the same type of supercapacitors can have different parameters, especially for the 5 F 2.7 V supercapacitors, whose R_2 value shows notable difference. Therefore, three samples are picked for the 5 F 2.7 V supercapacitor. This phenomenon may be due to the difference in the manufacture process.

With the increase of supercapacitor rated capacitance, the value of R_1 decreases, so does the value of R_2. The values of C_0 and K_v increases. This indicates the resistance of charge transfer decreases with the increasing of supercapacitor size, and the capacitance of the immediate branch increases with the supercapacitor size. For the delayed branch, the change of capacitance is more complicated.

It is shown that the value of K_c is very similar among different types of supercapacitors. Therefore, different types of supercapacitors share the similar charging/discharging characteristics. Although the variation of the K_r value is slightly larger, they still show similar rate of balancing effect between two branches among different size of supercapacitors. In general, the results indicate that the significance of charge redistribution phenomenon is similar among different size of supercapacitors.

However, it is to be noticed that the K_c value of 10 F supercapacitor is slightly smaller and the K_r value is larger. This indicates the charge redistribution phenomenon of the supercapacitors are less significant. This may be the result of previous extensive use of the 10 F 2.7 V supercapacitors. It is known that the long-term use could cause modification of the electrode structures and, particularly, a change in pore sizes and distribution [50]. Specifically, the diameters of the pores and the pore depth decrease with time. This is due to the impurities settlement around the wall of pores which reduces the surface area of activated carbon. Though the large pores can still accommodate electrons after power cyclings, many small pores are blocked or become too narrow for electrons. As a consequence, charge redistribution phenomenon become less significant since the storage capacity of the meso and micro pores decreases.

3.4.6 Supercapacitor Online State of Charge Prediction

The online state of charge prediction method combines the observer model and the power input model described before. More specifically, in order to accurately capture the dynamic behavior of supercapacitor, the proposed method predicts the future behavior of superca-

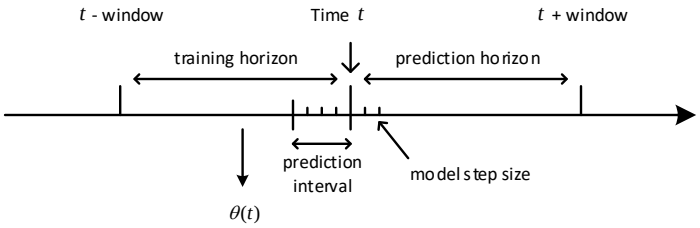


Figure 3.15: Illustration of the receding horizon mechanism

pacitor for a finite horizon based on the data in the previous horizon. The time span where prediction is carried out is called the prediction horizon while the time span used for calculating the supercapacitor parameters is called the training horizon. In the next prediction interval, the method is repeated for a shifted training horizon and prediction horizon, thus it operates in a receding horizon manner as illustrated in Figure 3.15. The time span of prediction interval is usually smaller than the that of prediction horizon. This way the training horizon of two consecutive runs have some overlap and the parameter calculated in the previous run can be used for initialization of next run. Since the parameters in the model reflects the physical characteristics of supercapacitor, which we assume don't have abrupt changes, the proposed method will be able to capture the slowly time varying terminal dynamics of supercapacitor.

Note that the parameter of the state observer is updated at each prediction interval according to the parameter determined in the training horizon. This way the observer always has an accurate estimation of the supercapacitor internal state. In each training horizon, given the measured charging/discharging power and the terminal voltage, an optimization problem similar to (3.63) can be formulated. $\mathrm{fitness}(\theta)$ represents the squared error between the measured voltage in the training horizon and the voltage estimated by the model using the parameter set θ. Moreover, the bound of the parameter search space can be specified as a neighborhood of the parameter value determined in the previous run to reduce the

computational cost.

In order to validate the proposed method, we conducted a dynamic charging/discharging experiment on a 310 F, 2.7 V supercapacitor using the MACCOR testing system. Before the test, the supercapacitor is charged by a 1 volt constant voltage source for one hour. Thus, the initial values of V_1 and V_2 are both 1 V. In the 7547 seconds dynamic test, two discharging pulses with relatively large magnitudes are implemented at time 2322 seconds and 4714 seconds respectively. The prediction interval of the proposed method is set to be 1 minute, and the prediction horizon is set to be 5 minutes.

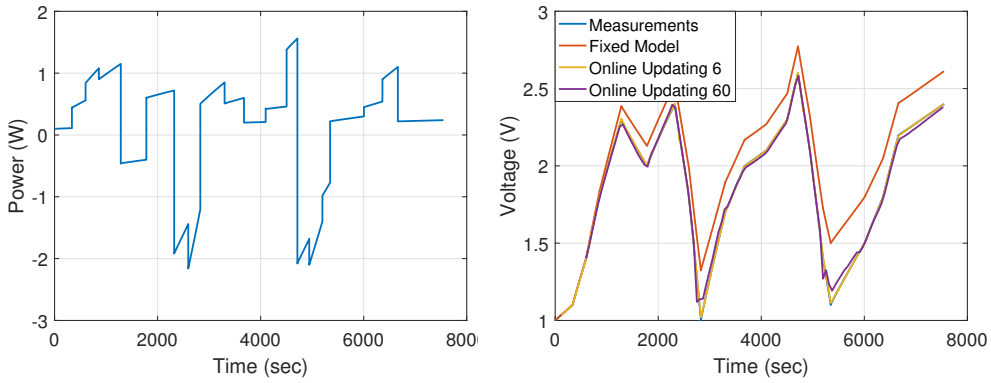
Figure 3.16 shows the prediction result of the proposed method. The upper panel is the charging/discharging power of the supercapacitor. The lower panel shows comparison of the measurements and the predicted terminal voltage. Three different prediction models are implemented for comparison. The first one is the predictor with fixed supercapacitor model parameters calculated with genetic algorithm (Table 3.4). The second one is the online predictor with an updating step size of 6 seconds. The prediction horizon is set to be 60 seconds. The third model is the online predictor with an updating step size of 1 minute and a prediction horizon of 5 minutes. It is demonstrated that the model with fixed parameters is not able to generate an accurate prediction within the experiment period. The online prediction method with 1 minute step size accurately captures the terminal voltage of supercapacitor despite the variation of charging/discharging power. The online prediction method with smaller step size provides better prediction performance, but at the cost of increased computational cost.

3.5 Adaptive Power Management Based on Model Predictive Control

3.5.1 System Dynamics

Based on the system model described before, the system dynamics includes the dynamics of data queue and the dynamics of energy storage device.

Consider the sensing, transmitting, receiving and data processing operations, the data



(a) (b)

Figure 3.16: Prediction results obtained with the proposed state of charge prediction method. (a) The charging/discharging power profile. (b) The comparison between the measured voltage and the prediction.

queue dynamics of a node can be described as

$$Q(t+1) = \left|Q(t) - f_t(t) - f_{dp}(t)\right|_+ + r_s(t) + f_r(t). \tag{3.77}$$

Moreover, the size of data queue is limited by the RAM size:

$$Q(t) \leq Q_{max}. \tag{3.78}$$

The transmitting and receiving rates are limited by the link capacity $c_r(t)$ and $c_t(t)$.

$$0 \leq f_r(t) \leq c_r(t), \tag{3.79}$$

$$0 \leq f_t(t) \leq c_t(t). \tag{3.80}$$

The data change rate $f_{dp}(t)$ is limited by the maximum number of data change within a time slot.

$$0 \leq f_{dp}(t) \leq c_{dp}(t). \tag{3.81}$$

The sensing, transmitting and receiving operations all consume energy. Their energy prices are $p_s(t)$, $p_t(t)$, and $p_r(t)$ respectively. The energy consumption of the data processing can be different for different applications. In this paper, we take the liner model as in (3.5) as an example, which corresponds to the average filtering process. Let $EC(t)$ be the amount of energy consumption at time slot t. It can be represented by

$$EC(t) = p_s(t)r_s(t) + EC_{dp}(t) + p_t(t)f_t(t) + p_r(t)f_r(t). \tag{3.82}$$

With the supercapacitor model depicted in (3.26), (3.30) and (3.57), the supercapacitor terminal behavior can be estimated. What is to be noted is that the terminal voltage of supercapacitor has to be

$$1 \leq V_{est} \leq 2.7, \tag{3.83}$$

to guarantee continuous operation.

3.5.2 Objective Function

The objective of the proposed power management method is to track a reference value r of the quality of service via controlling the sensing, transmitting, receiving and data processing operations of each node. According to Theorem 3.3.1, the average quality of service of all nodes in the network is the lower bound of the event detection probability of the network. Therefore, forming an optimization problem to track r ensures that the event detection probability of the network is around or greater than r.

The quality of service of each node could also be maximized to achieve the maximum event detection probability of the whole network. However, it leads to a higher sensing rate during the lifetime of the node, which is not efficient for energy conservation.

Therefore, in this paper we formulate the objective function of the optimization problem as

$$\underset{r_s, f_{dp}, f_t, f_r}{\text{minimize}} \sum_t (1 - \exp(-\lambda\alpha E_{QA}(t)\frac{r_s(t)}{r_{max}}) - r)^2. \tag{3.84}$$

3.5.3 Model Predictive Control

Based on the system dynamics and objective function presented above, the power management problem can be formulated as an optimization problem and solved with model predictive control. The model predictive control method has the advantage of explicitly handling the constraints and achieving optimal performance within a finite horizon. It also prevents the performance degradation caused by the uncertainty of communication scheduling.

The control inputs of the MPC are the sensing rate r_s, transmitting and receiving rates f_t and f_r and the data processing rate f_{dp}. The state variables are the data queue length $Q(t)$ and the internal states of supercapacitor $V_1(t)$ and $V_2(t)$. By adjusting r_s, f_t, f_r and f_{dp}, the objective function as in (3.84) can be minimized. The optimization formulation can be represented as

$$
\begin{aligned}
\underset{r_s, f_{dp}, f_t, f_r}{\text{minimize}} \quad & \sum_{t=1}^{k} (1 - \exp(-\lambda \alpha E_{QA}(t) \frac{r_s(t)}{r_{max}}) - r)^2 \\
\text{subject to} \quad & supercapacitor\ dynamics\ (3.26)(3.30)(3.57) \\
& data\ queue\ dynamics\ (3.77) \\
& 1 \leq V_{est}(t) \leq 2.7 \\
& 0 \leq Q(t) \leq Q_{max} \\
& 0 \leq r_s(t) \leq r_{max} \\
& 0 \leq f_{dp}(t) \leq c_{dp}(t) \\
& 0 \leq f_t(t) \leq c_t(t) \\
& 0 \leq f_r(t) \leq c_r(t)
\end{aligned}
\tag{3.85}
$$

In which, k represents the prediction horizon of the MPC.

This optimization problem is solved over a finite interval of k future time slots, which starts at the current time slot. Only the calculated control inputs corresponding to the first predicted time slot are actually applied to the system. The remaining control inputs are

discarded. At the next time slot, a new optimization problem with updated constraints is solved over a shifted prediction horizon. At each time slot, the control input applied to the system depends on the most recent measurements and the model deviation is minimized.

The particle swarm optimization (PSO) algorithm is employed here to solve the optimization problem because of its advantage of quick convergence and ability to efficiently handle nonlinearity. The PSO is a population based stochastic approach for solving optimization problems. The algorithm generates a group of random particles in the search space representing the candidate solutions to the optimization problem. Each particle searches for better solutions in the search space by adjusting its velocity based on two values. The first one is the best solution the particle has achieved so far, which can be represented as p^k_{best}. The second one is the best solution that has been obtained so far by all particles in the population, which we represent as g_{best}. The PSO algorithm can be summarized as Algorithm 4. In which, η_1 and η_2 represent the two random numbers.

Algorithm 2 Particle Swarm Optimization

Require: Number of particles N_p, number of iterations N_i

for k=1 : N_p **do**
  Initialize particle k;
end for
$i \leftarrow 1$;
for i=1 : N_i **do**
  for k=1 : N_p **do**
    Calculate fitness value of particle k;
    if Current solution is better than p^k_{best} **then**
      $p^k_{best} \leftarrow$ current solution;
    end if
    if p^k_{best} is better than g_{best} **then**
      $g_{best} \leftarrow p^k_{best}$
    end if
  end for
  for k=1 : N_p **do**
    $v_k = v_k + c_1 * \eta_1 * (p^k_{best} - x_k) + c_2 * \eta_2 * (g_{best} - x_k)$;
    $x_k = x_k + v_k$;
  end for
end for

Xu *et al.* [59] solve the MPC of a fast dynamic systems using the PSO implemented on a FPGA. For the power management problem formulated as (3.85), the control frequency can be much lower. Moreover, the trade-off of solution performance and the computational time can be explored to further reduce the computation time at each step. More specifically, due to the fast convergence of PSO, the generated solution could have satisfactory performance after a few generations. Thus the number of generations to be executed can be reduced depending on the time and energy budget. This usually won't cause significant performance degradation when system dynamics does not vary very fast, since only the solution corresponding to the next time slot is actually applied and the other solutions are discarded.

The search space of the PSO can also be constrained to a neighborhood of the solution calculated in the previous step. It not only reduces the computation time but also makes the calculated control command easier to implement. Furthermore, to accelerate the algorithm execution, each time when the optimization problem is solved, the calculated solution can be stored and used to initialize one of the particles for the next run.

On the other hand, for applications like radar based CPS, the nodes are usually more powerful than those used for low-power wireless sensor network. This makes implementing MPC on these sensor nodes possible.

3.6 Simulation Studies

To demonstrate the effectiveness of the proposed power management method, we simulate a randomly deployed radar network in which all nodes are powered by supercapacitors. Three simulation studies are designed. The first simulation study investigates a single node which is fully charged and the MPC is designed to track a reference QoS value. The second simulation study also focuses on a single node and the solar harvester is equipped to recharge the supercapacitor when there is solar energy available. The third simulation study implements a network of uniformly distributed radar nodes to detect a non-persistent

event randomly generated within the area of interest.

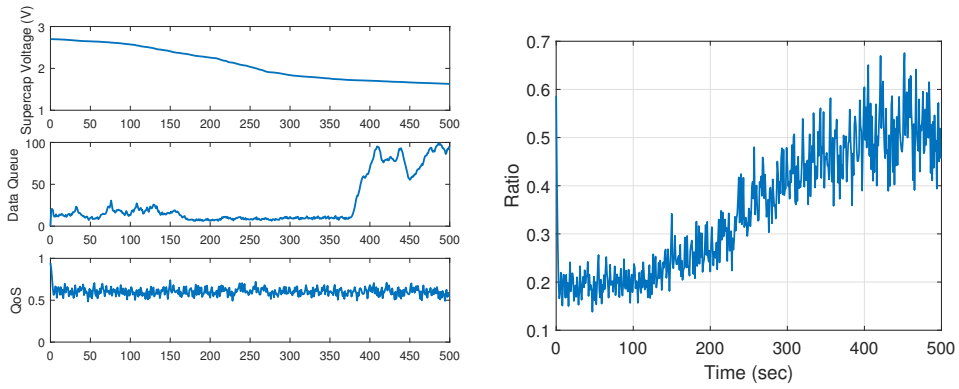
The supercapacitor state of charge online prediction method described in Section 3.46 is not implemented in this section, because it could operate on a completely different time scale in practice. Here we focus on investigating the proposed self-aware power management method. We assume each sensor is equipped with a fully charged 310 F 2.7 V supercapacitor with parameters listed as in Table 3.4.

3.6.1 Single Node No Harvesting

In this simulation study, assume the nodes are randomly distributed according to a spatial Poisson process with an intensity of $\lambda = 5$ and one of the nodes is investigated. The proportional coefficient between the radar footprint and the available energy is 0.00077858. The node is designed to track a reference QoS value of 0.6. For the radar sensor, the time varying sensing price was randomly generated using a uniform distribution over the range of $[0.008, 0.016]$ (J per packet) for each node at each time slot. The maximum sensing rate of the radar sensor is set to be 20 packets. In the MPC controller, we take p_s to be the average of the sensing price, which is 0.012 J per packet.

For the communication between nodes, the channel capacity c_{max} is set to be 40 packets. The receiving energy price is set to be 0.025 J per packet and the transmitting energy price is 0.015 J per packet. The ratio of the receiving energy price and transmitting energy price is set to be 0.6 according to the data sheet of a IEEE 802.15.4 transceiver (AT86RF231). The details of communication scheduling are simplified. We assume the number of the receiver nodes and transmitter nodes are randomly generated using a uniform distribution over the range of $[1, 4]$. In the MPC controller, we set p_t and p_r as the expectation of the transmitting and receiving energy price, which are 0.0375 and 0.0625 J per packet respectively. The power management algorithm then calculates the average number of packets sent to or received from the neighboring nodes.

The data processing process is assumed to be a simple averaging operation like (3.5)



(a) (b)

Figure 3.17: (a) The terminal voltage of the supercapacitor, the length of data queue, and the quality of service of each node. (b) The sensing ratio of r_s/r_{max}.

with $e_1 = 0.001\ J$ per packet and $e_2 = 0.0005\ J$ per packet. The maximum data size change that can be caused by data processing is $c_{dp} = 30$.

Initially, the supercapacitor is fully charged, i.e. the terminal voltage is 2.7 V and $V_1 = V_2 = 2.7$ V. The data queue is empty, so $Q(0) = 0$ packets. At each step, the MPC controller predicts the supercapacitor voltage within a finite horizon and solves an optimization problem to calculate the sensing rate of the radar sensor $r_s(t)$, the transmitting and receiving rates $f_t(t)$ and $f_r(t)$, and the data processing rate f_{dp}. The search space of the current solution is limited to $\pm 30\%$ of the solution found in the previous time step. The step size of the MPC is set to be 1 second and the prediction window is 5 steps.

Figure 3.17a shows the supercapacitor terminal voltage, size of data queue, and the node quality of service calculated by the proposed method over a 500 seconds simulation time. The quality of service is tracking a reference signal of 60% and the supercapacitor terminal voltage is within the range of $[1, 2.7]$ V. The length of data queue is within the range of $[0, 100]$ packets. Figure 3.17b is the result of the calculated sensing ratio r_s/r_{max} of radar sensor. It can be seen that the sensing ratio gradually increases to compensate for the decay of energy level. The size of data queue also shows slight increase, which is the result of decreasing transmitting rate so that the energy can be saved for sensing operations.

3.6.2 Single Node with Solar Harvester

In the second simulation study, the node is equipped with a solar harvester and the real world solar trace from [60] is used (Figure 3.18a). Assume the solar panel has a size of 5 cm $\times$ 5 cm. All sensor node parameters are the same as the previous study, except that the step size of the MPC is set to be 1 minute and the prediction window to be 5 minutes. Moreover, the WCMA algorithm [61] is implemented to predict the future harvesting solar energy. The WCMA algorithm is incorporated into the prediction model of the MPC along with the supercapacitor model.

The simulation is performed for a time period of 1 day and the simulation results are shown in Figure 3.18. It is demonstrated that despite the varying harvesting energy, the quality of service of the node keeps tracking the reference value of 0.6. The size of data queue and supercapacitor terminal voltage are both within the feasible ranges. It is worth noting that after 1500 seconds the system reaches a relatively stable status even though the sensing and communication process are stochastic and the harvested solar energy is varying.

In the first two simulation studies, it is demonstrated that the proposed method is effective for tracking a reference quality of service in the single node case. Since the PSO algorithm is a stochastic optimization technique and the solution at each step depends on the solution in the previous step, the calculated solution is therefore not necessarily optimal in many sense. However, the CPS system is stochastic in nature and pursuing optimality might not be realistic in many cases. The PSO based MPC method has many advantages in these applications. The optimality and computation time can be balanced by changing the iteration number of PSO. The system stochastic property is handled by the receding horizon fashion of MPC.

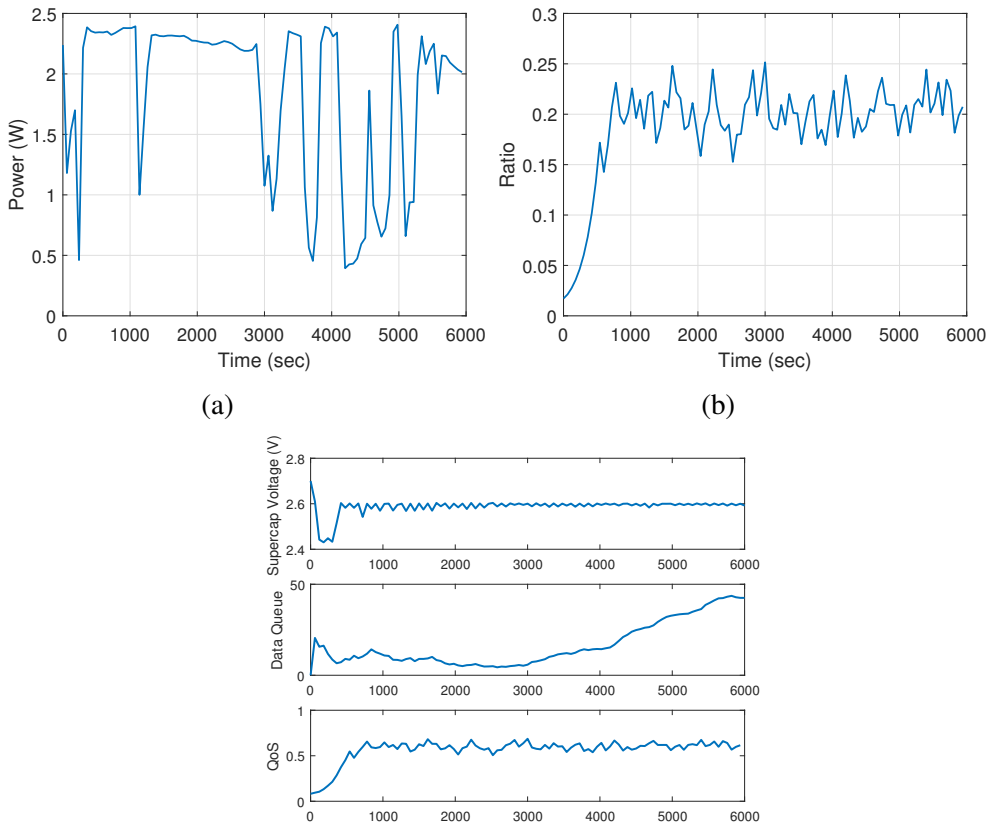


Figure 3.18: (a) The solar trace for energy harvesting. (b) The sensing ratio of r_s/r_{max}. (c) The terminal voltage of the supercapacitor, the length of data queue, and the quality of service of each sensor node.

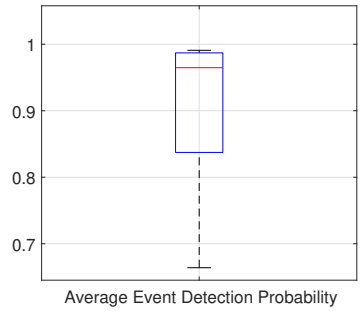


Figure 3.19: Boxplot of the average event detection probability

3.6.3 Radar Network for Event Detection

In this study, we investigate the proposed method under network circumstances. Each sensor node runs the proposed method and the overall performance of the network is evaluated via the detection probability of randomly generated events.

A Monte Carlo simulation is conducted in this section. We assume the nodes are randomly deployed within an unit area. The node deployment can be modeled as a spatial Poisson process with intensity of $\lambda = 10$. The ratio between the radar footprint and the available energy is set to be 0.005. No energy harvesting opportunity is assumed in this study.

10 simulation trials are conducted. Each trial generates a node sensor distribution. Then, at each time step, an event is generated at a random position within the area of interest. The event can be detected if it is within the footprint of at least one radar sensor and the sensor is on. So the probability of event detection can be calculated given the event position and the sensor distribution.

In each simulation trial, the sensor network is simulated for 240 seconds with a step size of 60 seconds. At each step, the controller for each sensor node is executed in sequence and the randomly generated event is evaluated in terms of detection probability. The average of event detection probability at all time steps are calculated for each trial, and the probability of all the trials are plotted in Figure 3.19 as box plot.

In all the simulation trials, the average event detection probability is greater than the reference quality of service value of 0.6. This accords with the Theorem 3.3.1. Therefore, it demonstrates that the controllers operating on the CPS nodes can guarantee that the overall event detection probability of the network is around or greater than the reference quality of service.

3.7 Summary

This part of dissertation work presents a power management method for supercapacitor-powered cyber-physical system based on the model predictive control and particle swarm optimization. Radar network is used as an example to demonstrate the proposed method. The event detection probability of the network is designed to maintain a satisfactory level, which is achieved by decomposing the network event detection probability as the quality of service of each node and each sensor node tracks a reference quality of service while satisfying the constraint of continuous operation. The varying charging/discharging characteristics of supercapacitor is captured by a proposed online state of charge prediction method that adapts to the change by updating the model parameters in a receding horizon fashion. The proposed method makes it possible to accurately predict the terminal behavior of the supercapacitor and make full use of the stored energy. Simulation studies are conducted to evaluate the proposed power management method and it is demonstrated that the proposed method can guarantee a satisfactory event detection probability for the sensor network while satisfying the operation constraints.

CHAPTER 4

ADAPTIVE POWER MANAGEMENT OF BIOIMPLANT

4.1 Introduction

Developing the power management method for bioimplant can be more complicated since the thermal effect of the implantable device is difficult to model.

The commonly used method to guarantee thermal safety of the implantable device is combining numerical method with in vivo experiments during the design phase [62, 63, 64]. The maximum power dissipation is determined based on the maximum tolerable temperature, which is then specified as a design constraint. However, the performance of the device is often limited due to the fact that an over-conservative design is chosen in many cases to account for the uncertainties of the actual operation.

Several researchers have conducted research related to the online thermal management. Communication scheduling methods are developed for biosensor network applications to prevent the temperature from increasing above the safe limit [65, 66, 67]. For example, a method for selecting the network leader that communicates with the base station is developed by Tang *et al.* [68] based on the leadership rotation history and the location of the next leader. These work only consider the overheating caused by communication and focuses on reducing overheating by communication scheduling. Wentz *et al.* [41] developed a wireless supercapacitor-based headborne device, which stimulates brain cells with an LED array. To solve the overheating issue, the LEDs are shutdown when the temperatures reach a predefined threshold. As a primitive method of thermal management, this method can not optimize the operation of the device. A similar approach is also employed in Luo *et al.* [69]. But instead of a temperature sensor, a thermoresponsive micro circuit breaker is used to protect the device from overheating. Moreover, Krishna *et al.* [70] provides a

comprehensive survey of the thermal management of cyber-physical systems. However, to the authors' knowledge, there have been few studies addressing the dynamic thermal management problem of the implantable device.

In this chapter, the black box based modeling technique is employed to predict the temperature increase in the surrounding tissue. The proposed thermal modeling method is then used to support adaptive power management.

4.2 Thermal Dynamics Analysis

This work start by studying the thermal effect of bioimplants. The heat transfer from the bioimplant to its surrounding tissue can be modeled using the famous Pennes bioheat equation [38]:

$$\rho C \frac{\partial T(\mathbf{x},t)}{\partial t} = \nabla \cdot (k \nabla T(\mathbf{x},t)) + A_0 - B_0(T(\mathbf{x},t) - T_b) + \rho SAR + P_{electronics}, \quad (4.1)$$

where $\mathbf{x}$ is the spatial coordinates, t is the time parameter, ρ and C refer to tissue density and specific heat; $\nabla \cdot (k \nabla T(\mathbf{x},t))$ models the thermal diffusion with k representing the thermal conductivity of tissue; $B_0(T(\mathbf{x},t) - T_b)$ models the effect of blood perfusion where T_b are the temperature of blood; and A_0 is the rate of the heat per unit volume of tissue produced by the source; ρSAR represents the heating effect due to SAR and $P_{electronics}$ is the power density of the external electronics.

A computational software COMSOL (Comsol Inc., Burlington, MA) that implements the FEA method is often used to solve the Pennes bioheat equation. The Utah electrode array (UEA) demonstrated in [40] is implemented, as shown in Figure 4.1. According to [40], the simulation results and experimental measurement are in good agreement. In this work, this experimentally validated numerical model is used in part to demonstrate the effectiveness of the proposed thermal management method. This method, however, is not suitable for predicting temperature increase in real-time.

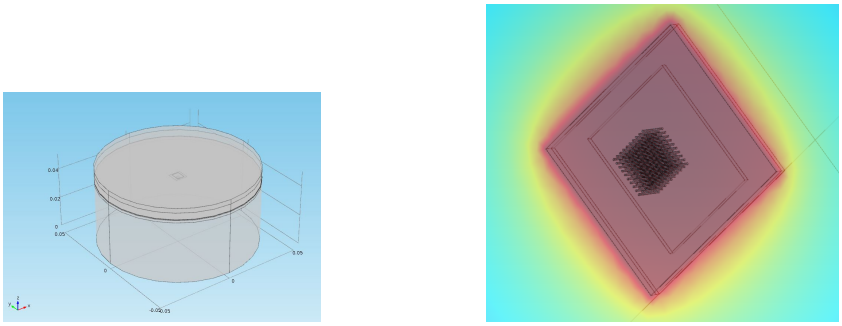


Figure 4.1: Illustration of the developed COMSOL model (a) cylinderical human brain model with height and radius being 50 mm. The thickness of the scalp and skull are 3 and 5 mm respectively. and (b) the UEA model with 10×10 micro-electrodes and a $7.88mm \times 7.53mm$ chip.

Numerous methods have been proposed to solve (4.1) more efficiently, including Finite Difference Time Domain (FDTD) methods [71], fourier transform based methods [72, 73], hybrid alternating-direction implicit (ADI) approach [74], and a method of superposition from separate sources combined with model simplification [75]. However, all these methods rely on sampling the temperature value in the simulation domain as they evolve in time. Their time complexity and space complexity make them unsuitable for real-time thermal management.

4.3 Simplified Thermal Model

To simplify the thermal model, the effect of blood perfusion and metabolism are neglected from (4.1) and modeled as disturbance. Moreover, the term $\nabla \cdot (k \nabla T(\mathbf{x}, t))$ in (4.1) can be replaced by $k \nabla^2 T(\mathbf{x}, t))$ for homogeneous materials, resulting in a second order parabolic partial differential equation:

$$\rho C \frac{\partial T(\mathbf{x}, t)}{\partial t} = k\left(\frac{\partial^2 T(\mathbf{x}, t)}{\partial x^2} + \frac{\partial^2 T(\mathbf{x}, t)}{\partial y^2} + \frac{\partial^2 T(\mathbf{x}, t)}{\partial z^2}\right) + P_{electronics}. \tag{4.2}$$

The heat flow described by this differential equation has a similar form as that for electrical current, and there is a well known duality between them. The simplified thermal

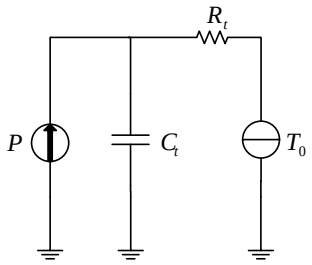


Figure 4.2: Simplified thermal model

model can be shown in Figure 4.2. More specifically, the heat flow (W) passing through a thermal resistor (°C/W) is equivalent to the electrical current (A) through an electrical resistance (Ohm), and the temperature difference (°C) corresponds to voltage difference (V). The heat absorbtion phenomenon can be denoted as the thermal equivalent capacitance (J/°C). The thermal conduction phenomenon can be represented as thermal equivalent resistance (K·m^2/W). The simplified thermal model is shown in [76].

Let P represent the power dissipation of the implantable device and T_0 represents the tissue temperature. R_t is the equivalent thermal resistance and C_t is the equivalent thermal capacitance. Then the temperature T of the tissue that has direct contact with the implantable device corresponds to the voltage of C_t, which can be calculated as

$$C_t \frac{dT}{dt} = \frac{T_0 - T}{R_t} + P. \tag{4.3}$$

The temperature can be solved numerically as

$$T(t) = (1 - \frac{\Delta t}{R_t C_t})T(t-1) + \frac{\Delta t}{R_t C_t} T_0 + \frac{\Delta t}{C_t} P(t-1). \tag{4.4}$$

In which, Δt is the step size.

Let $\Delta T(t) = T(t) - T_0$, then

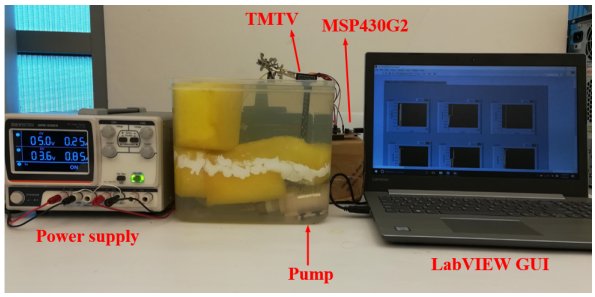
$$\Delta T(t) = (1 - \frac{\Delta t}{R_t C_t})\Delta T(t-1) + \frac{\Delta t}{C_t} P(t-1). \tag{4.5}$$

Therefore, the complex thermal dynamics of the implantable device can be approximated by a first order linear function. The effect of the metabolism, blood perfusion and electromagnetic field are considered as disturbance to the system. With the online multistep prediction method presented in Section 4.5, this linear function can capture the fundamental dynamics of thermal effect. If, for some cases, the metabolism, blood perfusion and electromagnetic field have more significant impact on the temperature increase in the body, a linear model with higher order could be used to model the thermal effect.

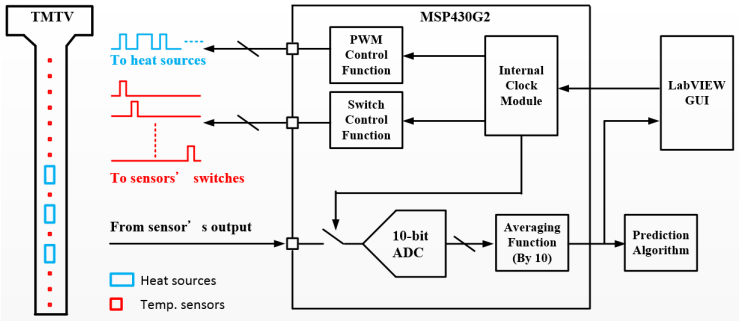
4.4 In-vitro Thermal Effect Test Vehicle

An in-vitro experiment system [77] is built to emulate the thermal effect of UEA. The system consists of three major components. The first one is a temperature monitoring and management test vehicle (TMTV) developed in the lab that has heat source and temperature sensors (TI LMT70) soldered on, which is used to emulate the implanted electronics. The temperature sensors have an accuracy around 0.1 °C and are also small in size. The second component is a water circulation system that uses a marine pump to control the flow rate and it is used to emulate the blood perfusion effect. Lastly, a monitor and control system is built with TI MSP430G2 board and Labview front end. The Labview front end on the PC is used to display and save the temperature measurements and it can also call the Matlab program which implements the thermal management algorithm. The TI MSP430G2 acts as the middleware between the TMTV and PC. It sends the control signal to the heat source on the TMTV and sends the temperature readings back to PC.

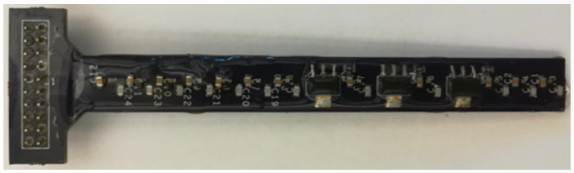
Figure 4.3 shows the developed hardware testing system. The container in the middle are filled with water and a marine pump is placed at the bottom to create water circulation



(a) The developed hardware testing system.



(b) Hardware diagram.



(c) The developed TMTV system. [77]

Figure 4.3: The in-vitro experiment system.

in the container. Note that the sponge material is also glued in the container to ensure a uniform water flow in the upper portion of the container where the TMTV is placed. To simulate the heat diffusion effect of blood perfusion, the water flow generated by the pump is adjusted to be similar to the blood perfusion rate in the human brain by choosing a pre-determined supply voltage.

This system provides accurate temperature measurements by calibrating each temperature sensor beforehand and applying Kalman filter to filter out the measurement noise. It can be used to evaluate the impact of different factors on the thermal dynamics and evaluate the performance of the proposed algorithm before conducting animal testing.

4.5 Online Multistep Prediction Method

A suitable model for generating accurate output predictions within a horizon is crucial to achieve high performance closed-loop control. The methods to obtain such models are often referred to as MPC relevant identification methods [78]. Compared to standard prediction error methods (PEM), MPC relevant identification methods often have better long term prediction performance.

Based on the iterative single step prediction error method (ISSPEM) proposed by Farina *et al.* [79], a recursive multi-step prediction error method (RMSPEM) is developed to generate the optimal prediction model for temperature increase within a horizon of k time steps. The method takes the power consumption and temperature measurements as input, then calculates the coefficients of the following linear temperature model (4.6) that minimizes the prediction error.

$$\Delta T(t) = a\Delta T(t-1) + bP(t-1). \tag{4.6}$$

The discrete time transfer function corresponding to (4.6) is

$$G(z) = \frac{bz^{-1}}{1 - az^{-1}}. \tag{4.7}$$

The numerator is denoted as $N_G(z) = bz^{-1}$. The denominator is denoted as $D_G(z) = 1 - az^{-1}$. $\theta = (a, b)^T$ represent the vector of model parameters.

4.5.1 Batch MSPEM

The multi-step prediction error (MSPEM) cost function and its optimization procedure for a batch of data are first derived herein. The MSPEM cost function represents the average prediction error within a horizon of k time steps.

Let $\Delta T(t+k|t)$ be the value of temperature increase predicted by iterating k times the

recursive equation of (4.6), as a function of $\Delta T(\cdot)$ up to time t and $P(\cdot)$ up to time $t+k$. It can be represented as [12]:

$$\Delta T(t+k|t) = R_k(z)\Delta T(t) + E_k(z)N_G(z)P(t+k) \tag{4.8}$$

where $R_k(z)$ and $E_k(z)$ can be calculated as:

$$R_k(z) = a^k \tag{4.9}$$

and

$$E_k(z) = \sum_{i=0}^{k-1} a^i z^{-i}. \tag{4.10}$$

Let

$$\phi_k(t) = \begin{pmatrix} \Delta T(t) \\ P(t+k-1) \\ \vdots \\ P(t) \end{pmatrix}, \tag{4.11}$$

and

$$\Theta_k(\theta) = \begin{pmatrix} a^k \\ b \\ ab \\ \vdots \\ a^{k-1}b \end{pmatrix}. \tag{4.12}$$

The k-step ahead predictor can be reformulated as:

$$\Delta T(t+k|t) = \phi_k(t)^T \Theta_k(\theta). \tag{4.13}$$

For a batch of N measurements of temperature increase and power consumption values,

define

$$Y_N^k = \begin{pmatrix} \Delta T(1+k) \\ \vdots \\ \Delta T(N) \end{pmatrix}, \tag{4.14}$$

and

$$\Phi_k = \begin{pmatrix} \phi_k(1)^T \\ \vdots \\ \phi_k(N-k)^T \end{pmatrix}. \tag{4.15}$$

The cost function of MSPEM is defined as

$$J_{MP}^N(k) = \frac{1}{k}\sum_{j=1}^{k} J_P^N(j), \tag{4.16}$$

in which

$$\begin{aligned} J_P^N(j) &= \frac{1}{N-j}\sum_{t=j+1}^{N}[\Delta T(t) - \Delta T(t|t-j)]^2 \\ &= \frac{1}{N-j}(Y_N^j - \Phi_j\Theta_j(\theta))^T(Y_N^j - \Phi_j\Theta_j(\theta)). \end{aligned} \tag{4.17}$$

This cost function can be minimized using the standard Newton method [12]. The optimal model parameters can be estimated iteratively as:

$$\begin{aligned} \theta_{j+1} = \theta_j - (\frac{\partial^2}{\partial\theta^2}J_{MP}^N(j)\Big|_{\theta=\theta_j})^{-1}\nabla_\theta J_{MP}^N(j)\Big|_{\theta=\theta_j}, \\ j = 1, \ldots, k-1. \end{aligned} \tag{4.18}$$

The Hessian of $J_{MP}^N(j)$ can be calculated as:

$$\frac{\partial^2}{\partial\theta^2}J_{MP}^N(j)\Big|_{\theta=\theta_j} = \frac{1}{j}\sum_{s=1}^{j}\frac{\partial^2}{\partial\theta^2}J_P^N(s), \tag{4.19}$$

and

$$\frac{\partial^2}{\partial\theta^2}J_P^N(s) \approx \frac{2}{N-s}\nabla_\theta\Theta_s(\theta)\Phi_s^T\Phi_s\nabla_\theta\Theta_s(\theta)^T. \tag{4.20}$$

For the sake of convenience, define

$$Q_N^s = \nabla_\theta\Theta_s(\theta)\Phi_s^T\Phi_s\nabla_\theta\Theta_s(\theta)^T. \tag{4.21}$$

The gradient of $J_{MP}^N(j)$ can be expressed as follows:

$$\nabla_\theta J_{MP}^N(j)\Big|_{\theta=\theta_j} = \frac{1}{j}\sum_{s=1}^{j}\nabla_\theta J_P^N(s) \tag{4.22}$$

and

$$\nabla_\theta J_P^N(s) = \frac{2}{N-s}\nabla_\theta\Theta_s(\theta)(\Phi_s^T\Phi_s\Theta_s(\theta) - \Phi_s^T Y_N^s). \tag{4.23}$$

In (4.70) and (4.72), Θ_j and its gradient $\nabla_\theta\Theta_j$ can be updated iteratively over j:

$$\Theta_{j+1} = W_j\Theta_j, \tag{4.24}$$

$$\nabla_\theta\Theta_{j+1} = \nabla_\theta\Theta_j {W_j}^T + ([1\ \mathbf{0}_{1,1+j}]\Theta_j){H_j}^T. \tag{4.25}$$

in which,

$$W_j = \begin{pmatrix} \begin{pmatrix} a \\ \mathbf{0}_{j,1} \\ b \end{pmatrix}, \begin{pmatrix} \mathbf{0}_{1,j+1} \\ I_{j+1} \\ \mathbf{0}_{1,j+1} \end{pmatrix} \end{pmatrix}, \tag{4.26}$$

and

$$H_j = \begin{pmatrix} \begin{pmatrix} I_1 \\ \mathbf{0}_{j+2,1} \end{pmatrix}, \begin{pmatrix} \mathbf{0}_{j+1,2} \\ I_2 \end{pmatrix} \end{pmatrix}. \tag{4.27}$$

And the initial value can be chosen as $\Theta_1 = \theta$, $\Delta_\theta\Theta_1 = I_2$.

4.5.2 RMSPEM

Then the optimization procedure of MSPEM is reformulated into a recursive form so that each time when a new temperature measurement and power consumption value are recorded, the prediction model could be updated accordingly.

Suppose the temperature measurements and the power consumption data are recorded sequentially. The Nth tuple contains the measurement of temperature increase and the power consumption at Nth discrete time instant.

Let

$$R_j^N = \Phi_j^T \Phi_j = \sum_{s=1}^{N-j} \phi_j(s)\phi_j(s)^T, \tag{4.28}$$

and

$$K_j^N = {Y_N^j}^T \Phi_j = \sum_{s=1}^{N-j} \Delta T(s+j)\phi_j(s)^T. \tag{4.29}$$

R_j^N can be calculated recursively as

$$R_j^N = R_j^{N-1} + \phi_j(N-j)\phi_j(N-j)^T. \tag{4.30}$$

On the right hand side of (4.80), R_j^{N-1} can be determined with all the data up to $N-1$th discrete time instant, and $\phi_j(N-j)$ contains all the data up to Nth discrete time instant.

Similarly, K_j^N can be calculated recursively as

$$K_j^N = K_j^{N-1} + \Delta T(N)\phi_j(N-j)^T. \tag{4.31}$$

In which, K_j^{N-1} can be determined with all the data up to $N-1$th discrete time instant. $\Delta T(N)$ is the measurement available at Nth discrete time instant. $\phi_j(N-j)$ requires the data up to Nth discrete time instant.

Furthermore, given the saved R_j^{N-1} and K_j^{N-1}, to calculate R_j^N and K_j^N, only the temperature measurements from time $N-k$ to time N and the power consumption values

from time $N-k$ to $N-1$ are needed. In practice, the temperature measurements and power consumption values could be saved in a FIFO queue of length $k+1$ and k respectively.

What is to be noticed is that, for $j = 1, \ldots, k$, the corresponding R matrix and K matrix have to be saved seperately. Each time, when a new tuple is recorded, R matrix and K matrix are updated using (4.80) and (4.31) for each j.

By using the recursive update of (4.80) and (4.31), the estimation of $\frac{\partial^2}{\partial \theta^2} J_P^N(s)$ and $\nabla_\theta J_P^N(s)$ can be reformulated as:

$$\frac{\partial^2}{\partial \theta^2} J_P^N(s) \approx \frac{2}{N-s} \nabla_\theta \Theta_s(\theta) R_s^N \nabla_\theta \Theta_s(\theta)^T, \tag{4.32}$$

$$\nabla_\theta J_P^N(s) = \frac{2}{N-s} \nabla_\theta \Theta_s(\theta) (R_s^N \Theta_s(\theta) - K_s^N). \tag{4.33}$$

Again, Θ_j and its gradient $\nabla_\theta \Theta_j$ can be updated using (4.73) and (4.74).

The algorithm procedure is illustrated in Algorithm 4.

Algorithm 3 Recursive MSPEM method

Require: Previously obtained parameter vector θ_{pre}, Input Queue U, Output Queue Y, R, K

$\theta = \theta_{pre}$;
$\Theta = \theta_{pre}$;
$\nabla_\theta \Theta = I_2$;
$\mu = 0.5$;
$Q_N = 0$;
$P_N = 0$;

1: **for** `j=1:Queue Length` **do**
2: Initialize ϕ_j;
3: $R[j] = R[j] + \phi_j \phi_j^T$;
4: $K[j] = K[j] + Y[end] \phi_j^T$;
5: $Q_N = \frac{1}{j}(Q_N + \nabla_\theta \Theta^T R[j] \nabla_\theta \Theta)$;
6: $P_N = \frac{1}{j}(P_N + \nabla_\theta \Theta^T (K[j]^T - R[j]\Theta))$;
7: $\theta = \theta + \mu {Q_N}^{-1} P_N$;
8: Calculate W_j, H_j;
9: $\nabla_\theta \Theta = W_j \nabla_\theta \Theta + H_j [1, \mathbf{0}_{1 \times j}] \Theta$;
10: $\Theta = W_j \Theta$;
11: **end for**

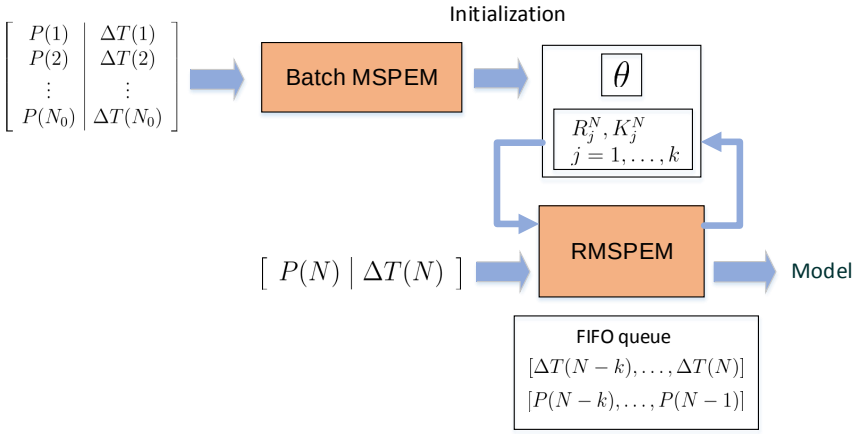


Figure 4.4: Illustration of the proposed method

4.5.3 Preprocessing

Note that the Newton method is only valid if matrix Q_N^j is nonsingular, $j = 1, \ldots, k$. This can be interpreted as a generalized identifiability condition.

For matrix Q_N^j to be full rank, it requires that

(a) $\nabla_\theta \Theta_j$ is full column rank (FCR),

(b) R_j^N is non-singular.

The first condition is investigated by Farian *et al.* [79]. As for the second condition, this requires the system input meets the excitation condition. This means, in practice, a preprocessing using the batch MSPEM is necessary before implementing the recursive MSPEM. The batch MSPEM is used to obtain the nonsingular matrix R_j^N for each j and the matrix K_j^N. These matrices along with the initial estimate of model parameter are then passed to the recursive MSPEM for online identification. The mechanism of the proposed method is shown in Fig 4.4.

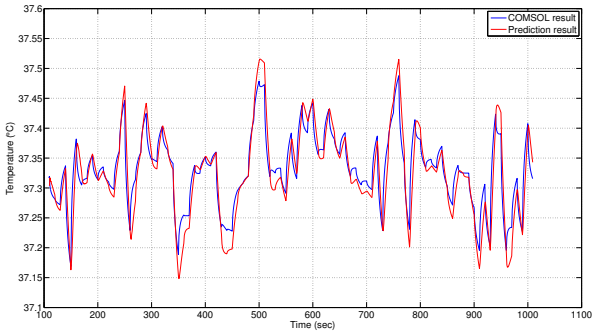


Figure 4.5: The prediction result of RMSPEM

4.5.4 Algorithm validation

To demonstrate the effectiveness of the proposed thermal modeling techniques, the results of the recursive MSPEM is compared with the results of COMSOL model described before. A probe used to measure the temperature is placed at $(x,\ y,\ z) = (0,\ 0,\ 0.042)$, which is the position below the heat source. The COMSOL simulation is conducted for 1000 seconds. The power dissipation value of the UEA at each second is generate randomly using a Gaussian distribution in order to satisfy the identifiability condition. The generated power dissipation value are also constrained within $[0, 0.02]\ mW$ to be realistic. The temperature measurements are recorded and converted into the temperature increase with respect to body temperature, then stored along with the generated power dissipation value at the same time instant.

For the RMSPEM, the prediction window is set to be 10 seconds. The first 100 seconds is used for the preprocessing. After the input and output queue are filled up at 110 seconds the recursive MSPEM update the parameters of the simplified thermal model according to the temperature increase obtained by COMSOL. Then the updated model is used to predict the temperature at 10 seconds later. This prediction is compared with the results obtained from COMSOL. The comparison results are shown in Figure 4.5. This comparison indicates that the thermal dynamics can be predicted by the linear model and RMSPEM with

relatively high accuracy. The Mean Square Error is about 8.04×10^{-4} °C. Thus this thermal modeling techniques can be used to obtain an accurate thermal model for the MPC.

Moreover, P in (4.6) represents the power consumption of the implantable devices, instead of the power dissipation. Generally, only part of the consumed power is dissipated as heat. This ratio can be determined by many factors. In practical applications, it may be difficult to track how much power is dissipated. One advantage of the proposed method is that this ratio can be learned adaptively according to the temperature feedback. Therefore, we can use the power consumption in (4.6) and regard the ratio as being already represented by coefficient b. Furthermore, the power consumption of embedded system is often controlled by adjusting the working mode of the device. Therefore, we can even use the controllable working mode to replace the power consumption P in (4.6) and the relation between the working mode and the power consumption can be learned online during the operation of the device. In general, by employing the simplified linear model and RMSPEM, more robust temperature control can be achieved, therefore enabling safe operation of implantable devices.

4.6 Adaptive Thermal Management Based on Online Multiple Step Prediction

4.6.1 System Description and Model

In the work of Wentz *et al.* [41], the system components of a neural prosthesis are described, which is relatively representative and therefore adopted for investigation in this paper. The neural prosthesis consists of optics module, radio module, power module, and motherboard module. The optics module holds up to 16 LEDs and is surgically affixed to the skull of a rat, while the remainder of the device is attached to the optics module via a low insertion force connector. Microelectrode array is another widely used tool for deep brain stimulation. In this dissertation, the implanted 3-D mciroelectrodes, specifically, the UEA, is used since its thermal effect has been studied in a previous literature [40] through experiment. The radio module mediates control command from a computer or laptop. The

motherboard module contains the microcontroller and the power circuitry. The power module includes a supercapacitor and an antenna for power reception. Since the temperature increase is caused by the power dissipated in the circuitry, the thermal effect can be investigated via modeling the power flow of the system. The whole hardware/software system is powered by the power received via the wireless interface. Here we use wireless interface to model the power antenna and radio chipset, both of which could potentially contribute to the heating of surrounding tissue. Energy storage device, such as battery or supercapacitor, is used to store the extra energy received by the system with some efficiency η. The software system and the controller are running on the microprocessor, which resides on the motherboard. It is also assumed that there is a temperature sensor connected to the motherboard, which is used to provide the feedback information for the thermal management method. The electrode array is implanted at the surface of the brain tissue. Depending on the placement of each component, microprocessor, wireless interface, and electrode array may contribute to the temperature increase in the surrounding tissue.

In the investigation, the power consumption of the microprocessor and electrode array can be adapted online. The power consumption of the electrode array can be adjusted by controlling the number of stimulation channels and the stimulation pulse train [40, 41]. The stimulation is, in many cases, achieved through current-mode stimulation pulse. The current intensity, the phase and interphase durations, as well as the stimulation frequency are the parameters related to the power consumption. Furthermore, the performance of each component is proportional to its power consumption. For the micro-electrode array, the number of channels and the pulse train impact the device performance, such as the visual characteristics (diameter, depth, brightness, and duration) in the case of retinal prosthesis [80]. The power consumption of the components must also be greater than certain threshold to guarantee minimum operation. The transmitted power is also assumed to be adjustable. The designed controller adapts the working status of each component to optimize the device operation based on the information like task requirement, temperature readings and the

energy level of the storage device.

4.6.2 Power Management with MPC

To develop the power management framework for implantable device, the system state space model needs to be built based on the application requirements. The system state variables often include all the factors that are related to the system performance. For example, as in [57], the system state variables can include the temperature increase with respect to body temperature and the energy level of the storage device. In this thesis, we assume that the battery is adopted as the storage device and the model in [81] is modified to obtain the battery state space model.

Based on [81], let the battery state variables be $x = x_1$ and $y = x_0 + x_1$, the discrete-time state space equation can be modified as

$$\begin{pmatrix} x(t) \\ y(t) \end{pmatrix} = \begin{pmatrix} 1 - \lambda_1 \Delta t & 0 \\ -\lambda_1 \Delta t & 1 \end{pmatrix} \begin{pmatrix} x(t-1) \\ y(t-1) \end{pmatrix} + \begin{pmatrix} \frac{2}{\alpha} \\ \frac{3}{\alpha} \end{pmatrix} i(t-1), \tag{4.34}$$

in which $i(t-1) = \frac{P(t-1)}{V_t(t-1)}$ represents the current input, V_t is the terminal voltage of the battery and P represents the output power. $y(t)$ is defined to be 0 when the battery is at full capacity and 1 when the battery is dead. In practice, the terminal voltage of the battery can be measured using an ADC pin of the microprocessor. The reason for this modification is that the the extended quadratic programming problem with the state variables in the whole horizon can be solved very efficiently according to [82].

The control input of the system are the power consumption of the UEA and the microcontroller, and the received power of wireless interface. They are represented as P_{uea}, P_{mcu}, and P_{coil} respectively. By adjusting the control input, the state of charge of the battery can be controlled. From the thermal perspective, the power consumption of UEA is the only factor directly related to the heating of the implantable device. In practical applications, the control input can be chosen as the adaptable system parameters that have direct relation

to the power consumption of the components. The objective function of neural prosthesis operation can be defined based on the application requirements. For example, the objective can be tracking an optimal energy level r. Assume the optimal working mode of a component has a power consumption of P_r and its power consumption at time t is $P(t)$, then $\sum_{t=1}^{N}(P(t) - P_r)^2$ can be added to the objective function to maximize the component performance.

The system state space model is specified as the constraint of the optimization problem. So is the thermal safety constraint. A maximum allowable temperature ΔT_{max} can be added as a constraint to prevent the hot spot from heating too much. The power consumption of the microprocessor and the UEA must not exceed the maximum allowable value. The received power must be less than a certain threshold, which ensures there is no safety issues. The optimization problem is solved over a horizon of future k time steps, which starts at current time t. But only the control input corresponding to t in the solution sequence is actually applied to the system. The remaining control inputs are discarded. At the next time step, a new optimization problem based on the new measured data is solved over a shifted prediction horizon. At every time step, the control input applied to the system depends on the most recent measurements.

To prevent the effect of external disturbance and model deviation caused by the simplified thermal model, the linear thermal model is updated at every time instance. The RMSPEM algorithm calculates the optimal coefficient for the linear model given the temperature measurements and the control inputs in previous time steps. However, the model update doesn't have to be executed at each time step. A possible way to reduce the computational cost is to call the RMSPEM only when the error of the predicted temperature and the actual measurement becomes larger than a certain threshold.

4.6.3 Simulation Studies

The developed thermal management method is investigated with two simulation studies of the neural prosthesis. The COMSOL model developed in Section 4.2 was used to emulate the thermal effect of the neural prosthesis. It was then used in conjunction with Matlab to perform real-time simulation. We implemented MPC for exemplary case studies using the CVX, a package for specifying and solving convex programs [83].

Adaptive Thermal Management

The first simulation study was designed to emulate the thermal effect of a wireless neural prosthesis that doesn't have an energy storage device. The inductive coil and microprocessor are planted on the surface of the scalp. Only the UEA is implanted in the brain tissue, which is connected to other parts of the system through wire. The prosthesis stimulates different layers of the brain using the UEA, whose performance is proportional to its power consumption during normal operation. The UEA has a power consumption within the range between $0.04\ Watts$ and $0.065\ Watts$ during active operation, and its power consumption is assumed to be $0\ Watts$ while being put into the sleep mode. The maximum allowable temperature of the surrounding tissue is set to be 0.9 °C above the body temperature.

The following MPC is designed to maximize the device performance and maintain safe operation via regulating the power consumption of the UEA. In this example, the thermal safety is achieved by adding a constraint that limits the maximum allowable temperature. The performance optimization is achieved by maximizing the power consumption of UEA.

The quadratic program of the MPC can be formulated as

$$\begin{aligned} \min \quad & \sum_{t=1}^{k} (P_{max\_uea} - P_{uea})^2 \\ \text{s. t.} \quad & \Delta T(t) = a\Delta T(t-1) + bP_{uea}(t) \\ & \Delta T(t) \leq \Delta T_{max} \\ & 0 \leq P_{uea} \leq P_{max\_uea} \end{aligned} \tag{4.35}$$

In which, P_{uea} denotes the power consumption of the UEA and P_{max_uea} is its maximum value. $\Delta T(t)$ represents the temperature increase with respect to body temperature. ΔT_{max} is the maximum allowable temperature increase. The receding horizon for the MPC is chosen to be 10 time steps. At each time step, after the quadratic program is solved, P_{uea} is set to be 0 if the calculated value is less than 0.04 W, since the minimum operational power consumption of UEA is 0.04 W.

To investigate the performance of the developed method, three different cases were designed. Three different cases were designed to investigate the performance of the developed method. The first case uses a method adopted in a previously published system [41] to control the operation of the UEA. With the method, the UEA is set to the maximum working mode at the beginning and turned to the sleep mode once the temperature reaches 37.9 °C. After the temperature drops below 37.1 °C, the UEA is then set back to the maximum working mode. It repeats this process during the operation, so the UEA is either on or off according to the temperature measurements. The second case uses the batch MSPEM to identify the thermal model in advance and then uses the MPC to calculate the desired power consumption at run time. The third case combines the batch MSPEM preprocessing with the RMSPEM and adjusts the thermal model parameter based on the real-time temperature measurements. Results of these cases indicates all three cases are able to maintain safe operation.

The three cases are evaluated in terms of operation time and the square of difference

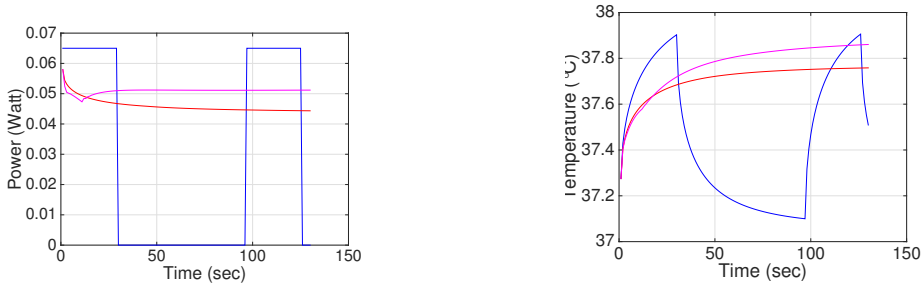


Figure 4.6: (a) UEA power consumption of all three cases. (b) Temperature measurement of all three cases. The blue curve represents the Case 1. The red curve represents the Case 2. The magenta curve represents the Case 3.

between the calculated power consumption and the maximum power consumption. The results are shown in Table 4.1. The operation time measures the total time when the power consumption of the UEA is greater than 0.04 W. The operation time of the three cases are 58 seconds, 130 seconds, and 130 seconds respectively. The square of difference with respect to the maximum power consumption are 0.3042, 0.40475, and 0.0259 respectively. Moreover, to quantify the complexity of the RMSPEM, the total execution time of RMSPEM for 130 seconds is recorded, which is 0.2593 seconds on a desktop computer with an i7-3770 CPU and 16 GB of RAM.

Table 4.1: Simulation results of adaptive thermal management

Simulation Cases	Safe Operation	Operation Time (s)	Square difference (W^2)
Case 1 On/off two states	Yes	58	0.3042
Case 2 Batch pre-processing	Yes	130	0.40475
Case 3 Batch pre-processing and online update	Yes	130	0.0259

It is shown that by dynamically adjusting the working status of the UEA, the operation time can be significantly prolonged. The cases of MPC with real-time updating and the one with only batch pre-processing all maintain operation during the entire simulation time. Moreover, MPC with RMSPEM updating (Case 3) shows better performance compared to MPC with batch MSPEM pre-processing (Case 2). This is because the accuracy of the prediction model is significantly improved by adjusting model parameter based on the real-time measurements.

Joint Thermal and Power Management

The second simulation study adds the energy storage device into the prediction model and studies the power management of a battery powered neural prosthesis. The neural prosthesis is assumed to have the same layout as in the previous study. The battery has a charging efficiency of 0.8. The optimal energy level of the battery is assumed to be 0.5 for the task. The controller tracks the optimal energy level of the battery by adjusting the received power of the inductive coil P_{coil}, the power consumption of the main board P_{main} and the power consumption of the UEA P_{uea}. Moreover, noise is added to the measured output. The battery level measurements noise is generated according to an autoregressive autoregressive with exogenous (ARARX) [84] input model.

The goal of the power management is to achieve sustainable operation, performance maximization, and thermal safety. The sustainable operation and performance maximization are guaranteed by designing the objective function of the quadratic programming. The thermal safety during operation is ensured by adding a constraint that limits the maximum temperature. The designed quadratic programming can be formulated as

$$\begin{aligned}
\text{min} \quad & \sum_{t=1}^{k}[(y(t)-r)^2+\gamma(P_{max_uea}-P_{uea})^2] \\
\text{s. t.} \quad & Battery\ equation\ (4.34) \\
& Temperature\ equation\ (4.6) \\
& 0 \leq y(t) \leq 1 \\
& \Delta T(t) \leq \Delta T_{max} \\
& P_{min_main} \leq P_{main} \leq P_{max_main} \\
& P_{min_coil} \leq P_{coil} \leq P_{max_coil} \\
& 0 \leq P_{uea} \leq P_{max_uea}
\end{aligned} \tag{4.36}$$

In which, r represents the optimal energy level and γ is the relative weight.

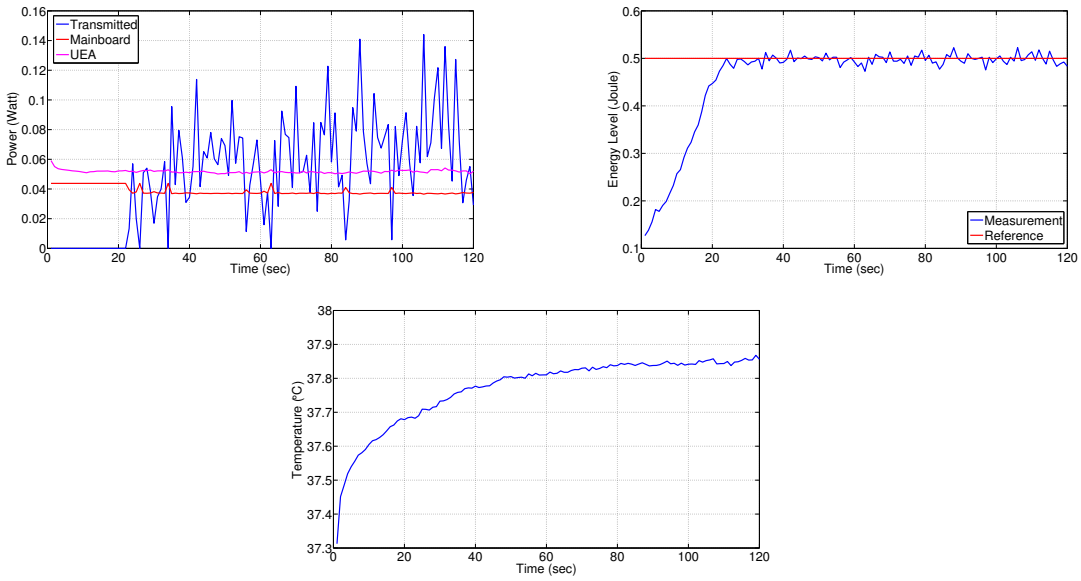


Figure 4.7: (a) The power consumption of each component. (b) The battery energy level. (c) Temperature of the surrounding tissue.

The receding horizon is chosen to be 10 time steps. As previously stated, P_{uea} is set to be 0 when the solution is less than 0.04 W. The battery terminal voltage is assumed to be a constant value 3.3 V given the fact that it varies little when there is no significant change in the state of charge. The battery parameter α is set to be 2.8, and λ is set to be 0.04.

The results of the power management are shown in Figure 4.7. The power consumption of the mainboard, the UEA and the received power from inductive coil are all within the specified range. The power consumption of the UEA are maximized to improve the performance of neural prosthesis. The energy level of the battery tracks the reference value of 0.5 despite the measurement noise. The temperature of the surrounding tissue is below the safe threshold 37.9 °C during the entire simulation time, so the thermal safety constraint is satisfied.

4.7 Bayesian Recursive Multistep Prediction Method

To further improve the prediction accuracy and increase the step size of the previous method, a Bayesian recursive multi-step prediction error method (Bayesian RMSPEM)

is developed. In this section, this prediction method and the adaptive thermal management method built based on the method are introduced.

4.7.1 System Model and Identification Criterion

Let's assume the underlying data-generating system $\mathcal{F}$ is defined as a general Single Input Single Output (SISO) linear discrete-time equation of the type:

$$y(t) = G(q, \theta^o)u(t) + H(q, \theta^o)e(t), \tag{4.37}$$

where the true parameters of the system are denoted by θ^o. $G(q, \theta^o)$ represents the transfer function from input to output, and $H(q, \theta^o)$ is the transfer function from a white noise source e to output additive disturbances. Both $G(q, \theta^o)$ and $H(q, \theta^o)$ are asymptotically stable transfer functions. q denotes the shift operator $qy(t) = y(t + 1)$.

Depending on how to parameterize G and H, many model structures have been proposed, such as the autoregressive with exogenous terms (ARX) model, the autoregressive-moving average with exogenous terms (ARMAX) model, and the Box-Jenkins model. In this paper, the estimation model with the output error (OE) structure are considered.

Let $G(q, \theta^o)$ be defined as

$$G(q, \theta^o) = \frac{N_G(q)}{D_G(q)} = \frac{b_1^o q^{-1} + \cdots + b_{n_b^o}^o q^{-n_b^o}}{1 + a_1^o q^{-1} + \cdots + a_{n_a^o}^o q^{-n_a^o}}, \tag{4.38}$$

and

$$\theta^o(\mathbf{a}^o, \mathbf{b}^o) = \begin{bmatrix} \mathbf{a}^o \\ \mathbf{b}^o \end{bmatrix} = \begin{bmatrix} -a_{n_a^o}^o \\ \vdots \\ -a_1^o \\ b_1^o \\ \vdots \\ b_{n_b^o}^o \end{bmatrix} \tag{4.39}$$

represents the system parameter in G. The numerator $N_G(z)$ and the denominator $D_G(z)$ are assumed to be coprime.

The OE prediction model can be represented as

$$\hat{y}(t|\theta) = \hat{G}(q,\theta)u(t), \tag{4.40}$$

which doesn't explicitly model the noise. In (4.40),

$$\hat{G}(q,\theta) = \frac{\hat{N}_G(q)}{\hat{D}_G(q)} = \frac{b_1 q^{-1} + \cdots + b_{n_b} q^{-n_b}}{1 + a_1 q^{-1} + \cdots + a_{n_a} q^{-n_a}}, \tag{4.41}$$

and

$$\theta = \begin{bmatrix} \mathbf{a} \\ \mathbf{b} \end{bmatrix} = \begin{bmatrix} -a_{n_a} \\ \vdots \\ -a_1 \\ b_1 \\ \vdots \\ b_{n_b} \end{bmatrix}. \tag{4.42}$$

In general, $n_a \neq n_a^o$ and $n_b \neq n_b^o$.

Suppose the input output data of $\mathcal{F}$ are recorded in sequential in the time domain as

$$\mathcal{L} = \{u(1), y(1), \ldots, u(N), y(N)\}. \tag{4.43}$$

Then the criterion of fit can be defined for the j-step-ahead prediction as

$$J_P^N(j) = \frac{1}{N - n_b - j + 1} \sum_{t=n_b+j}^{N} [y(t) - \hat{y}(t|t-j)]^2 + \gamma \theta^T P^{-1} \theta. \tag{4.44}$$

In which, $\hat{y}(t|t-j)$ denotes the prediction of $y(t)$ given the output data up to $t-j$ and input data up to t. Prior information of the system parameters are taken into account by

introducing a regularization term $\theta^T P^{-1}\theta$ into (4.44) with P^{-1} represents the covariance information of the parameter prior distribution. γ denotes the relative weight of the regularization term.

The multi-step prediction error criterion is defined as the average of the j-step-ahead criterion with $j = 1, \ldots, k$:

$$J_{MP}^N = \frac{1}{k}\sum_{j=1}^{k} J_P^N(j). \tag{4.45}$$

Gianluigi Pillonetto *et al.* [85, 86] summarized the kernel options for system identification, such as Diagonal/Correlated (DC) kernel, Tuned/Correlated (TC) kernel and Stable Spline (SS) kernel, whose frequency properties are summarized in [87]. The kernels can also be derived for system identification purposes [88]. The regularized cost function can then be solved by the regularized least square estimation.

4.7.2 Regularized Batch Pre-processing

To determine the hyperparameters of the regularization and choose a good starting point for searching the minimum of the cost function presented in the previous section, a batch of data is used to initialize the Bayesian RMSPEM algorithm. This batch of data is called the pre-processing data and represented as $\mathcal{L}^o = \{u^o(1), y^o(1), \ldots, u^o(N_0), y^o(N_0)\}$. In practice, this procedure helps to generate a reliable model estimation in the initial phase. In this section, we present how the regularization technique can be used for the multi-step prediction under the batch setting.

First, (4.40) can be converted into the linear regression form as

$$\hat{y}(t|\theta) = \phi(t)^T\theta. \tag{4.46}$$

Given the data set $\mathcal{L}^o$, the one-step predictions can be concatenated into the vector form

$$Y = \Phi\theta, \tag{4.47}$$

in which

$$Y = \begin{bmatrix} \hat{y}(1|\theta) \\ \vdots \\ \hat{y}(N_0|\theta) \end{bmatrix}, \tag{4.48}$$

and

$$\Phi = \begin{bmatrix} \phi(1)^T \\ \vdots \\ \phi(N_0)^T \end{bmatrix}. \tag{4.49}$$

Let the regularization matrix P be parameterized in terms of the hyperparameter η. The hyperparameter can be determined through

$$\hat{\eta} = \arg\min_{\eta} \ Y^T Z(\eta)^{-1} Y + \log|Z(\eta)|, \tag{4.50}$$

$$Z(\eta) = \Phi P(\eta) \Phi^T + \gamma^2 I_{N_0}, \tag{4.51}$$

which represents the maximization of the negative log likelihood function for estimating η from Y.

Let's then derive the multi-step prediction error cost function and its optimization procedure for a batch of data. Let $\hat{y}(t+j|t)$ be the output value predicted by iterating j times the recursive equation of (4.40). It can be represented as [12, 79]:

$$\hat{y}(t+j|t) = R_j(q) y(t) + E_j(q) \hat{N}_G(q) u(t+j), \tag{4.52}$$

where $R_j(q)$ and $E_j(q)$ can be calculated as

$$R_j(q) = \mathbf{C}\mathbf{A}^j \begin{bmatrix} q^{-n_a+1} \\ \vdots \\ 1 \end{bmatrix}, \tag{4.53}$$

and

$$E_j(q) = \mathbf{C}\sum_{i=0}^{j-1}\mathbf{A}^i\mathbf{B}q^{-i}. \tag{4.54}$$

In which,

$$\mathbf{A} = \begin{bmatrix} 0 & 1 & \dots & 0 \\ \vdots & \vdots & \ddots & \\ 0 & 0 & \dots & 1 \\ -a_{n_a} & -a_{n_a-1} & \dots & -a_1 \end{bmatrix}, \tag{4.55}$$

$$\mathbf{B} = \mathbf{C}^T, \tag{4.56}$$

and

$$\mathbf{C} = \begin{bmatrix} 0 & 0 & \dots & 1 \end{bmatrix}. \tag{4.57}$$

Let

$$Y_t = \begin{bmatrix} y^o(t-n_a+1) \\ \vdots \\ y^o(t) \end{bmatrix}, \tag{4.58}$$

and

$$U_{t,j} = \begin{bmatrix} u^o(t+j-1) \\ u^o(t+j-2) \\ \vdots \\ u^o(t-n_b+1) \end{bmatrix}. \tag{4.59}$$

Then define

$$\phi_j(t) = \begin{bmatrix} Y_t \\ U_{t,j} \end{bmatrix}, \tag{4.60}$$

and

$$\Theta_j(\theta) = \begin{bmatrix} (\mathbf{C}\mathbf{A}^j)^T \\ M_{j,n_b}\mathbf{b} \end{bmatrix}, \tag{4.61}$$

in which

$$M_{j,n_b} = \begin{bmatrix} \mathbf{CB} & \dots & \mathbf{CA}^{j-1}\mathbf{B} & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & \dots & 0 & \mathbf{CB} & \dots & \mathbf{CA}^{j-1}\mathbf{B} \end{bmatrix}^T. \tag{4.62}$$

The j-step-ahead predictor can be reformulated as a linear regression form:

$$\hat{y}(t+j|t) = \phi_j(t)^T \Theta_j(\theta). \tag{4.63}$$

$\Theta_j(\theta)$ is the j-step-ahead mapping of the predictor parameter θ.

For the batch pre-processing data, define

$$Y_{N_0}^j = \begin{bmatrix} y^o(n_b+j) \\ \vdots \\ y^o(N_0) \end{bmatrix}, \tag{4.64}$$

and

$$\Phi_j = \begin{bmatrix} \phi_j(n_b)^T \\ \vdots \\ \phi_j(N_0-j)^T \end{bmatrix}. \tag{4.65}$$

Let $R_j^0 = \Phi_j^T \Phi_j$ and $K_j^0 = \Phi_j^T Y_{N_0}^j$.

The cost function of batch pre-processing is

$$J_{MP}^{N_0}(k) = \frac{1}{k} \sum_{j=1}^{k} J_P^{N_0}(j), \tag{4.66}$$

in which

$$J_P^{N_0}(j) = \frac{1}{N_0 - n_b - j + 1} \|Y_{N_0}^j - \Phi_j \Theta_j(\theta)\|^2 + \gamma \theta^T P^{-1} \theta. \tag{4.67}$$

This cost function can be minimized using the standard Newton method. The optimal

model parameters can be estimated iteratively as:

$$\theta_{j+1} = \theta_j - (\frac{\partial^2}{\partial\theta^2} J_{MP}^{N_0}(j)\Big|_{\theta=\theta_j})^{-1} \nabla_\theta J_{MP}^{N_0}(j)\Big|_{\theta=\theta_j}, \quad j = 1, \ldots, k-1. \tag{4.68}$$

The Hessian of $J_{MP}^{N_0}(j)$ can be calculated as:

$$\frac{\partial^2}{\partial\theta^2} J_{MP}^{N_0}(j)\Big|_{\theta=\theta_j} = \frac{1}{j}\sum_{s=1}^{j} \frac{\partial^2}{\partial\theta^2} J_P^{N_0}(s), \tag{4.69}$$

and

$$\frac{\partial^2}{\partial\theta^2} J_P^{N_0}(s) \approx \frac{2}{N_0 - n_b - s + 1} \nabla_\theta \Theta_s(\theta) R_s^0 \nabla_\theta \Theta_s(\theta)^T + 2\gamma P^{-1}. \tag{4.70}$$

The gradient of $J_{MP}^{N_0}(j)$ can be expressed as follows:

$$\nabla_\theta J_{MP}^{N_0}(j)\Big|_{\theta=\theta_j} = \frac{1}{j}\sum_{s=1}^{j} \nabla_\theta J_P^{N_0}(s) \tag{4.71}$$

and

$$\nabla_\theta J_P^{N_0}(s) = \frac{2}{N_0 - n_b - s + 1} \nabla_\theta \Theta_s(\theta)(R_s^0 \Theta_s(\theta) - K_s^0) + 2\gamma P^{-1}\theta. \tag{4.72}$$

In (4.70) and (4.72), Θ_j and its gradient $\nabla_\theta \Theta_j$ can be updated iteratively over j:

$$\Theta_{j+1} = W_j \Theta_j, \tag{4.73}$$

$$\nabla_\theta \Theta_{j+1} = \nabla_\theta \Theta_j W_j^T + ([\mathbf{C}\ \mathbf{0}_{1,n_b+j-1}]\Theta_j) H_j^T. \tag{4.74}$$

in which,

$$W_j = \begin{bmatrix} \begin{bmatrix} \mathbf{A}^T \\ \mathbf{0}_{j,n_a} \\ \mathbf{bC} \end{bmatrix}, \begin{bmatrix} \mathbf{0}_{n_a,n_b-1+j} \\ I_{n_b-1+j} \\ \mathbf{0}_{1,n_b-1+j} \end{bmatrix} \end{bmatrix}, \tag{4.75}$$

and

$$H_j = \begin{bmatrix} \begin{bmatrix} I_{n_a} \\ \mathbf{0}_{n_b+j,n_a} \end{bmatrix}, \begin{bmatrix} \mathbf{0}_{n_a+j,n_b} \\ I_{n_b} \end{bmatrix} \end{bmatrix}. \tag{4.76}$$

The initial value can be set as $\Theta_1 = \theta$, $\nabla_\theta \Theta_1 = I_{n_a+n_b}$.

During the pre-processing, R_s^0 and K_s^0 are stored for $s = 1, \ldots, k$ and used for initialization of the RMSPEM. The calculated θ also provides an initial start to accelerate the convergence of RMSPEM. In practical applications, the model order selection procedure can be incorporated into the pre-processing. The model order that minimizes the cost function (4.66) can be used for the following Bayesian RMSPEM. This saves the computational cost during the run time. Due to the robustness to model order, many system variations during the run time can still be captured. Moreover, as will be demonstrated below, the computational cost of the Bayesian RMSPEM is proportional to the model order. In many cases, a low model order can be generated in the pre-processing procedure, which helps to reduce the complexity of the proposed algorithm.

4.7.3 Bayesian Recursive MSPEM

After the regularization hyperparameters and the RMSPEM initial parameters R_s^0, K_s^0 and θ_0 are determined through the pre-processing, the Bayesian RMSPEM algorithm updates the parameter estimation at each time instant when there is new data available. Assume the test data is represented as $\mathcal{L} = \{u(1), y(1), \ldots, u(N), y(N), \ldots\}$, the update procedure of the proposed algorithm will be demonstrated by assuming that the current time instant is N, and the new available data are the input $u(N)$ and output $y(N)$. The model parameters are then updated iteratively using the Newton method based on the prediction error of $y(N)$.

As shown in the batch pre-processing, to determine the Hessian matrix and the gradient of the cost function, the R matrix and K matrix must be calculated for $j = 1, \ldots, k$. Let's define

$$R_j^N = R_j^0 + \Phi_j^T \Phi_j \tag{4.77}$$

and

$$K_j^N = K_j^0 + {Y_N^j}^T \Phi_j. \tag{4.78}$$

At time N, R_j^N and K_j^N must be stored for $j = 1, \ldots, k$.

R_j^N can be represented as

$$R_j^N = R_j^0 + \sum_{s=n_b}^{N-j} \phi_j(s)\phi_j(s)^T, \tag{4.79}$$

which can then be calculated recursively as

$$R_j^N = R_j^{N-1} + \phi_j(N-j)\phi_j(N-j)^T. \tag{4.80}$$

On the right hand side of (4.80), R_j^{N-1} can be determined with all the data up to $(N-1)$th discrete time instant, and $\phi_j(N-j)$ contains all the data up to Nth discrete time instant.

Similarly, K_j^N can be represented as

$$K_j^N = K_j^0 + \sum_{s=n_b}^{N-j} \phi_j(s)y(s+j), \tag{4.81}$$

and it can be calculated recursively as

$$K_j^N = K_j^{N-1} + \phi_j(N-j)y(N). \tag{4.82}$$

In which, K_j^{N-1} can be determined with all the data up to $(N-1)$th discrete time instant. $y(N)$ is the output measurement available at Nth discrete time instant. $\phi_j(N-j)$ requires the data up to Nth discrete time instant.

Furthermore, given the saved R_j^{N-1} and K_j^{N-1}, it only requires the input and output values within a finite time window to calculate R_j^N and K_j^N. More specifically, it requires the output measurements from time $N-k-n_a+1$ to time $N-1$ and input values from time

$N-k-n_b+1$ to time $N-1$ to calculate $\phi_j(N-j)$, $j=1,\ldots,k$. To calculate K_j^N, the newest output measurement $y(N)$ is also needed. In practice, all the output measurements and input values necessary to update the R matrix and K matrix can be saved in a FIFO queue as

$$\begin{aligned} Input\ Queue: & \begin{bmatrix} u(N-k-n_b+1) \\ \vdots \\ u(N-1) \end{bmatrix}, \\ Output\ Queue: & \begin{bmatrix} y(N-k-n_a+1) \\ \vdots \\ y(N) \end{bmatrix}. \end{aligned} \tag{4.83}$$

For each $j=1,\ldots,k$, the corresponding R matrix and K matrix have to be saved separately. Each time, when the new input and output values are recorded, R matrix and K matrix are to be updated using (4.80) and (4.82) for each j.

With the recursive updates of (4.80) and (4.82), the estimation of $\frac{\partial^2}{\partial\theta^2}J_P^N(s)$ and $\nabla_\theta J_P^N(s)$ can be formulated as:

$$\frac{\partial^2}{\partial\theta^2}J_P^N(s) \approx \frac{2}{N-n_b-s+1}\nabla_\theta\Theta_s(\theta)R_s^N\nabla_\theta\Theta_s(\theta)^T + 2\gamma P^{-1}, \tag{4.84}$$

$$\nabla_\theta J_P^N(s) = \frac{2}{N-n_b-s+1}\nabla_\theta\Theta_s(\theta)(R_s^N\Theta_s(\theta) - K_s^N) + 2\gamma P^{-1}\theta. \tag{4.85}$$

Again, Θ_j and its gradient $\nabla_\theta\Theta_j$ can be updated using (4.75) and (4.76). Let

$$Q_N^j = \frac{1}{j}\sum_{s=1}^{j}\nabla_\theta\Theta_s(\theta)R_s^N\nabla_\theta\Theta_s(\theta)^T + 2\gamma P^{-1}, \tag{4.86}$$

$$-P_N^j = \frac{1}{j}\sum_{s=1}^{j}\nabla_\theta\Theta_s(\theta)(R_s^N\Theta_s(\theta) - K_s^N) + 2\gamma P^{-1}\theta. \tag{4.87}$$

Then Q_N^j and P_N^j can be calculated iteratively for $j = 1, \ldots, k$ as:

$$Q_N^j = \frac{j-1}{j} Q_N^{j-1} + \frac{1}{j} \nabla_\theta \Theta_j(\theta) R_N^j \nabla_\theta \Theta_j(\theta)^T + \frac{2}{j} \gamma P^{-1}, \tag{4.88}$$

$$P_N^j = \frac{j-1}{j} P_N^{j-1} + \frac{1}{j} \nabla_\theta \Theta_j(\theta)(K_s^N - R_s^N \Theta_s(\theta)) - \frac{2}{j} \gamma P^{-1} \theta. \tag{4.89}$$

The parameter update procedure can be represented as

$$\theta_{j+1} = \theta_j + \mu (Q_N^j)^{-1} P_N^j. \tag{4.90}$$

The computation procedure of the proposed method is summarized in Algorithm 4.

Algorithm 4 Bayesian RMSPEM method

Require: Previously obtained parameter θ_{pre}, Input Queue U, Output Queue Y, R, K

$\theta = \theta_{pre}$;
$\Theta = \theta_{pre}$;
$\nabla_\theta \Theta = I_{n_a+n_b}$;
Set μ;
$Q_N = 0$;
$P_N = 0$;

1: **for** `j=1:k` **do**
2: Initialize ϕ_j;
3: $R[j] = R[j] + \phi_j \phi_j^T$;
4: $K[j] = K[j] + \phi_j Y[k]$;
5: $Q_N = \frac{j-1}{j} Q_N + \frac{1}{j} \nabla_\theta \Theta R[j] \nabla_\theta \Theta^T + \frac{2}{j} \gamma P^{-1}$;
6: $P_N = \frac{j-1}{j} P_N + \frac{1}{j} \nabla_\theta \Theta (K[j] - R[j]\Theta) - \frac{2}{j} \gamma P^{-1} \theta$;
7: $\theta = \theta + \mu {Q_N}^{-1} P_N$;
8: Calculate W_j, H_j;
9: $\nabla_\theta \Theta = \nabla_\theta \Theta W_j^T + ([C, \mathbf{0}_{1 \times n_b+j-1}]\Theta) H_j^T$;
10: $\Theta = W_j \Theta$;
11: **end for**

4.7.4 Algorithm Extension and Analysis

Forgetting Factor

For identification of time-varying systems, the aforementioned method can be modified so that past data become less relevant for the current estimation. In this subsection, we propose a routine that use the forgetting factor to weight the past data.

Following a classical practice in parametric time-varying system identification [89], we introduce a forgetting factor $\lambda \in (0, 1]$ into the update procedure in order to base the estimation mainly on the more recent data. Specifically, we modify the j-step-ahead cost function to be

$$J_P^N(j) = \frac{1}{N - n_b - j + 1} \|\Lambda_N^j (Y_N^j - \Phi_j \Theta_j(\theta))\|^2 + \gamma \theta^T P^{-1} \theta, \tag{4.91}$$

in which,

$$\Lambda_N^j = \begin{bmatrix} \lambda^{\frac{N-n_b-j}{2}} & & & \\ & \lambda^{\frac{N-n_b-j-1}{2}} & & \\ & & \ddots & \\ & & & \lambda^0 \end{bmatrix} \tag{4.92}$$

and λ can often chose from 0.98 to 0.995. By using this forgetting factor, measurements older than $T_0 = \frac{1}{1-\lambda}$ samples are included in the criterion with a weight that is $e^{-1} \approx 36\%$ of that of the most recent measurement.

With the modified cost function, the algorithm update procedure remains the same while the update of data matrix R_s^N and K_s^N in Hessian matrix (4.88) and Gradient (4.89) can be modified as

$$\bar{R}_j^N = \lambda \bar{R}_j^{N-1} + \phi_j(N-j)\phi_j(N-j)^T, \tag{4.93}$$

$$\bar{K}_j^N = \lambda \bar{K}_j^{N-1} + \phi_j(N-j)y(N). \tag{4.94}$$

Performance Analysis

The time complexity of Algorithm 4 can be analyzed in terms of the number of flops (floating-point operation). For each $j = 1, \ldots, k$, the calculation requires order $(n_a + n_b)^3$ and j^2 flops. Therefore, the entire algorithm for $j = 1, \ldots, k$ requires order $(n_a + n_b)^3$ and k^3. More specifically, the time complexity is in an order of magnitude similar to square matrix multiplication. In real applications, benefiting from the property of robustness to model orders, the computational cost can be reduced with a lower model order and shorter prediction range.

In terms of the space complexity, besides the input queue of length $k + n_b - 1$ and the output queue of length $k + n_a$, the algorithm needs to store $R[j]$ and $K[j]$ for each $j = 1, \ldots, k$. $R[j]$ is a matrix in $\mathcal{R}^{(n_a+n_b+j-1)\times(n_a+n_b+j-1)}$ and $K[j]$ is a vector in $\mathcal{R}^{n_a+n_b+j-1}$. The previous parameter estimation $\theta \in \mathcal{R}^{n_a+n_b+j-1}$ also needs to be stored.

Online system identification methods like RPEM are often less computationally demanding. However it cannot guarantee long term prediction performance, especially in the case where complex noise models are involved. Therefore, it is not very suitable for practical applications like adaptive MPC, where the prediction accuracy within a certain horizon is crucial to the performance of the closed-loop control.

The method proposed in [90] has higher computational cost compared to the method proposed in this paper, as it maintains a high order model and rely on hyper-parameter updating at each execution to select the appropriate model. The hyper-parameter calculation process is both computationally demanding and difficult to implement for embedded systems, like what is used in the bioimplants. Moreover, the method in [90] requires a sampling rate that is several times of the model updating rate, which is also a challenge for many applications.

Compared to the online identification techniques likes RPEM, the developed algorithm falls between the online identification and the batch identification. It uses the preprocessing to determine the kernel hyperparameters and initialize RMSPEM. In applica-

tions where the prediction accuracy is crucial to the performance, this prevents the bad performance in the initial phase of the algorithm. During the operation, the developed method has the advantage of low complexity and robustness to different noise models. Even when the predetermined model structure is underparameterized, the developed method still captures the low-frequency fundamental dynamics. With the forgetting factor incorporated, the developed method is able to track a time varying system and provide a k-step-ahead prediction based on the history information within the k prior steps.

4.7.5 Simulation Investigation

In this section, the properties of the developed method are demonstrated with three simulation studies. The first simulation study is a Monte Carlo test with underparameterized prediction models, wherein the order of the prediction model is lower than that of the data generation system. The second simulation study is a Monte Carlo test that features different noise models. The third simulation study demonstrates the performance of the developed method with a linear time varying system.

In these simulation studies, the system generates two kinds of data sets. The first type is the pre-processing data set $\mathcal{L}^o = \{u^o(1), y^o(1), \ldots, u^o(N_0), y^o(N_0)\}$. The second type is the test data set $\mathcal{L} = \{u(1), y(1), \ldots, u(N), y(N)\}$.

Underparameterized Model

A Monte Carlo study of 100 runs is considered regarding identification of discrete-time OE models (4.40). At each run, a different 30th order transfer function is generated using the procedure described in [85]. A second order input filter is also generated using the similar procedure.

The input in the pre-processing data set $\mathcal{L}^o$ is the realization from white Gaussian noise of unit variance filtered by the input filter. The delay of the input is equal to 1. Starting from zero initial conditions, 1000 input-output data are collected with the output corrupted

by an additive white Gaussian noise. The signal-to-noise ratio (SNR) is randomly chosen with in $[1, 10]$ at every run. In the left two test cases of Figure 4.8, the preprocessing data set contains the first 150 input-output data while all the 1000 data are used in the right two cases.

Two types of test data sets are generated at every run. The first one contains the white noise corrupted output obtained using a unit variance white Gaussian noise as input. The second one is obtained with a test input generated using the same procedure as in the pre-processing data.

The performance measure (4.95) as in [85], which represents the variance of the prediction model, is adopted in this paper to compare different estimated models. The prediction horizon is chosen to be 20 steps.

$$\mathcal{F}_k(\hat{\theta}) = 100\left(1 - \sqrt{\frac{\sum_{t=k+1}^{N}(y(t) - \hat{y}(t|t-k))^2}{\sum_{t=k+1}^{N}(y(t) - \bar{y})}}\right). \tag{4.95}$$

The following 7 estimation methods are implemented for comparison:

1. *RecursiveARX*: It implements the recursive PEM approach with ARX model of 8th order. The estimated model is used to predict the output of 20 steps ahead. The estimator is implemented with the *rarx* Matlab routine.

2. *RecursiveOE*: The recursive OE estimator implements the OE model of 20th order and it predicts the output of 20 steps ahead according to the new available data.

3. *RMSPEM+CV*: The RMSPEM algorithm with model order selected via cross validation (CV). Specifically, the pre-processing data are split into two parts $\mathcal{L}_a^o$ and $\mathcal{L}_b^o$, containing the first and last $\frac{N}{2}$ input-output pairs in $\mathcal{L}^o$ respectively. The candidate models have the structure that the polynomials B and F have the same order which varies between 1 and 30. For OE models with different orders, the model parameters are obtained by the batch pre-processing with the estimation data $\mathcal{L}_a^o$. Then

the prediction errors are computed for the validation data $\mathcal{L}_b^o$. The model order that maximizing the prediction performance is selected and the final model parameter estimation is calculated with batch pre-processing for the complete data set $\mathcal{L}^o$.

4. *RMSPEM+Or*: The RMSPEM algorithm with an oracle (Or). In particular, for different model orders between 1 and 30, we use the batch pre-procesing to calculate the model parameters with $\mathcal{L}^o$. Then the oracle chooses the model structure that maximizes the fit on the test data. It represents a case that is impractical in general but provides a reference for performance evaluation.

5. *RMSPEM*: The RMSPEM algorithm that uses the OE model of 20th order.

6. *RMSPEM+DC,TC,SS*: The Bayesian RMSPEM algorithm equipped with DC, TC, and SS respectively. The employed model is 20th order. During the pre-processing, the kernel hyperparameters are estimated by solving the marginal likelihood optimization.

Figure 4.8 shows the boxplots of the 100 performance measures calculated in the Monte-Carlo study. The left panels are the results that use only the first 150 input output data for preprocessing, and the right panels are the results that use full 1000 input output data during preprocessing. The top panels show the performance measures with the white input signal and the bottom panels are the performance with the filtered input signal like in the pre-processing data set. The vertical axis represent the performance measure for each estimator.

In all the four simulation cases, the *Recursive ARX* method achieves good performance, but is not as good when the input is filtered.The PEM method can't guarantee the k-step-ahead prediction accuracy. The *RMSPEM+CV* approach has good prediction performance for the case with filtered input signal, but for white input signal the performance is unacceptable, especially when there is less pre-processing data. The *RMSPEM+Or* represents the ideal case where the test data is available for determining the model structure. It is

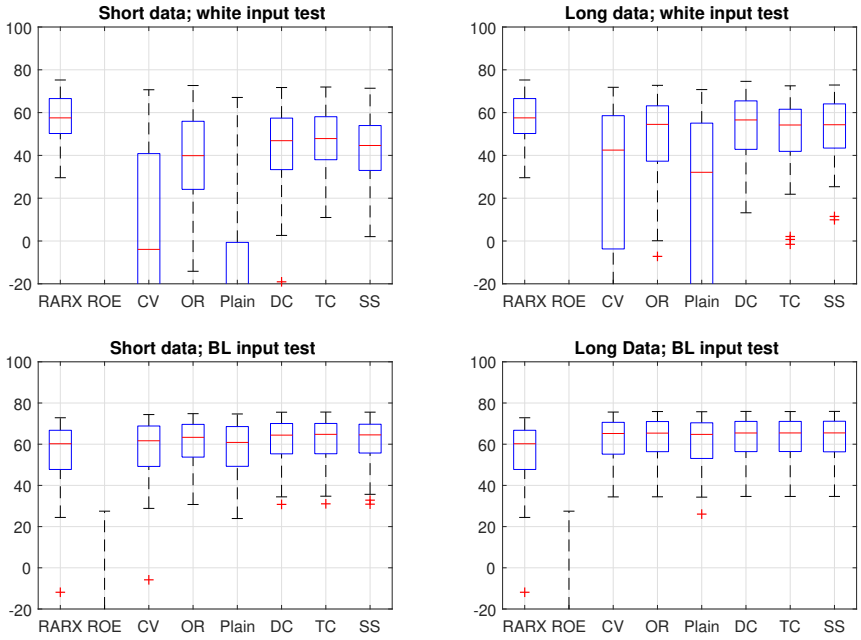


Figure 4.8: Identification of discrete-time OE-models.

shown that *RMSPEM+Or* achieves good prediction performance in all of the four scenarios and is therefore used as reference. The *RMSPEM* algorithm without any regulator is also implemented. When the input signal is white noise, the prediction performance is significantly inferior compared to other estimators. With the incorporated kernels, the Bayesian RMSPEM achieves satisfactory prediction for 20 steps ahead. In the white noise input case, the Bayesian RMSPEM even outperforms the *RMSPEM+OR*.

Additionally, it is obvious that if the data used for pre-processing is similar to the test data, the prediction performance is generally better. This is because the initial estimate obtained through the preprocessing is more likely to be in the neighborhood of the "good" estimate. More preprocessing data helps to improve the prediction performance, but in the Bayesian RMSPEM case the improvement is limited. Therefore, it is shown that the developed Bayesian RMSPEM method is very robust to the pre-processing data.

Box-Jenkins System

Let's consider a Box-Jenkins type data-generation system, in which $G(q) = \frac{B(q)}{F(q)}$ and $H(q) = \frac{C(q)}{D(q)}$ are 30th order transfer functions generated using the procedure described in the previous section. The SNR is randomly chosen from 1 to 10. The system is excited by two types of input signal. The first type of input signal is a white Gaussian noise with unit variance. The second type is the realization from white Gaussian noise filtered by a second order filter. The delay of the input is equal to 1. Starting from zero initial conditions, 1000 input-output data are collected.

Two sets of pre-processing data are generated. Both contains the first 200 input output data of the system excited with filtered white Gaussian noise. The first preprocessing data set is generated using only the system process model $G(q)$. The second preprocessing data set is generated using both the process model $G(q)$ and the noise model $H(q)$ in the Box-Jenkins model. Moreover, two sets of test data are used, which include one having the input with the same characteristic of the pre-processing data and another one that use white Gaussian noise as input.

The following estimators are used in this study:

1. *RecursiveARX*: It implements the recursive PEM approach with ARX model of 30th order. The estimated model is used to predict the output 20 steps ahead.

2. *RecursiveOE*: The recursive OE estimator implements the OE model of 30th order. The prediction horizon is 20 steps.

3. *RMSPEM+CV, RMSPEM+Or, RMSPEM*: These three estimators use the same setup as in the previous simulation study.

4. *RMSPEM+DC,TC,SS*: The Bayesian RMSPEM approach with DC, TC, SS kernels respectively. The kernel hyperparameters and the weight γ are determined using the preprocessing data.

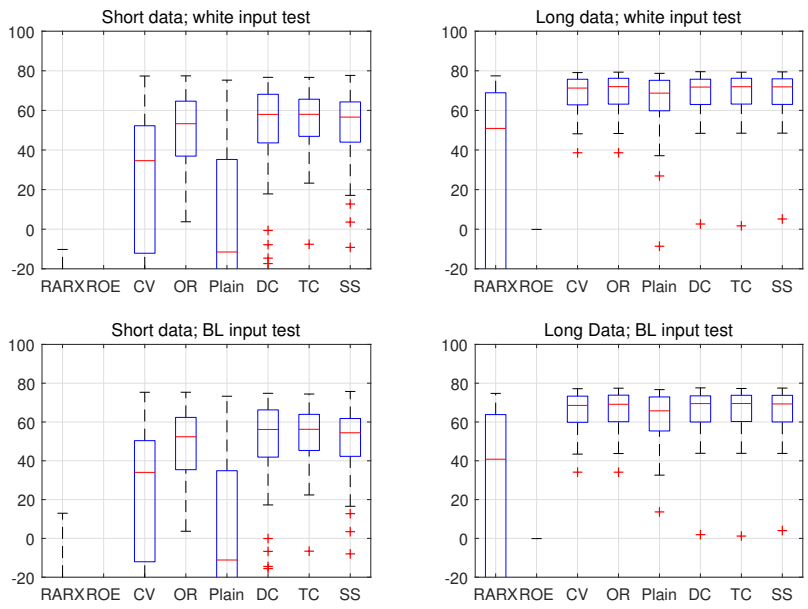


Figure 4.9: Identification of discrete-time Box Jenkins models.

The Monte Carlo study runs 80 tests. The prediction performance are plotted in Figure 4.9.

It is demonstrated in this study that the proposed method generally has a superior performance over the Recursive ARX method despite the type of kernel used. Moreover, generating pre-processing data with only the process model $G(q)$ gives a better initial estimation, thus the Bayesian RMSPEM better captures the underlying process model in the Monte Carlo tests shown in the left panel of Figure 4.9.

LTV System

In this study, the online parameter identification of linear time varying system with unknown order is investigated. The plant has two operating modes. The first mode has a 30th order transfer function generated randomly using a similar process as described in Section 4.7.5. The transfer function of the second mode is generated by perturbing the transfer function of the first mode with two additional poles and zeros. Thus both order and param-

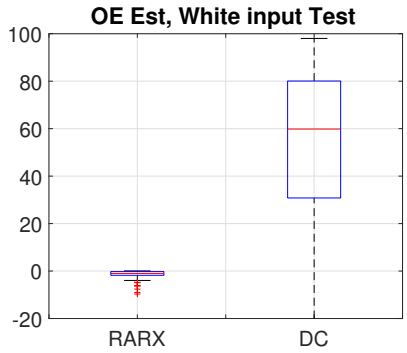


Figure 4.10: Identification of linear time varying system.

eters of the time varying system change when switching from the first mode to the second mode. 100 data sets consisting of 3000 input-output measurement pairs are generated using Monte Carlo simulations. The system switch at time $k = 1001$. The input of the system is generated as the realization of a unit variance Gaussian signal filtered by a randomly generated second order filter.

The first 400 input output data are used for pre-processing and the rest of them are used for testing. Two estimators are implemented for comparison. The first one is the recursive ARX method which chooses the model order that minimizes the prediction error at each time instant. The second one is the Bayesian RMSPEM method with DC regularizer whose hyperparameters are determined during the preprocessing process. Both recursive ARX and the Bayesian RMSPEM use a forgetting factor of 0.98. The prediction window is set to be 20 steps and the order of the prediction model is chosen to be 30.

The prediction performance of the Monte Carlo study is shown in Figure 4.10 with the y axis representing the performance measure calculated using (4.95). It is shown that the proposed method is able to track the switch of the time varying system, while the recursive ARX method fails to do so. Moreover, Bayesian RMSPEM is considerably faster than the recursive ARX with order selection. As is shown in Table 4.2,for the 3000 input-output data, the mean cumulative time of the two estimators are 0.0506 seconds and 1.1755 seconds respectively.

Table 4.2: Execution time of the two estimators

	ARX order_select	Bayesian RMSPEM
mean	1.1755 s	0.0506 s
std	11.7552	0.5002

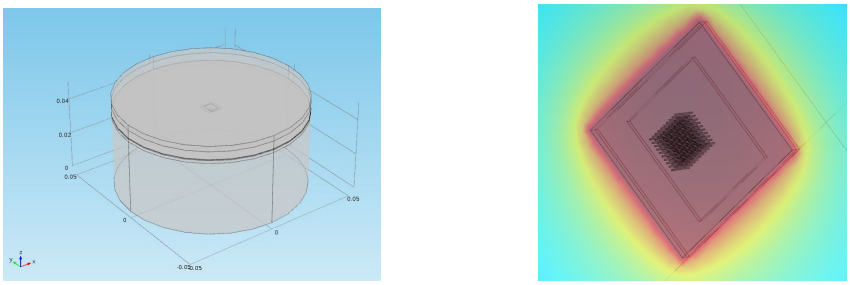
4.7.6 Thermal Modeling of UEA

In this section, the proposed Bayesian RMSPEM method is employed to predict the thermal effect of Utah electrode array. The proposed method is suitable for this application is because it provides an accurate temperature prediction with low computational complexity. The performance of the method is demonstrated with a COMSOL simulation and an in-vitro experiment.

COMSOL simulation

In this study, the developed method is used to model the thermal effect of the UEA. A COMSOL Multiphysics model (Figure 4.1) is implemented for what is demonstrated in [40]. The UEA is placed on the surface of the brain tissue and a probe is place at $(x, y, z) = (0, 0, 0.042)$ to measure the temperature. The COMSOL simulation is conducted for 1000 seconds. The power dissipation of the UEA is randomly generated every 10 seconds using a Gaussian distribution, which are then constrained within $[0, 0.02]$ mW. The temperature measurements are recorded and converted into the temperature increase with respect to the body temperature, then stored along with the generated power dissipation at the same time instant.

Bayesian RMSPEM is used to generate a model that predicts the temperature increase of the UEA given its power dissipation. The prediction window of the Bayesian RMSPEM is set to be 10 steps. Each step is 10 seconds. The data of first 200 seconds are used for the pre-processing. The Bayesian RMSPEM updates the parameters of a 5th order prediction model according to the temperature increase obtained by COMSOL. Then the updated model is used to predict the temperature 10 steps later via the j-step-ahead predictor (4.63).



(a) cylinderical human brain model. (b) the UEA model.

Figure 4.11: Illustration of the developed COMSOL model

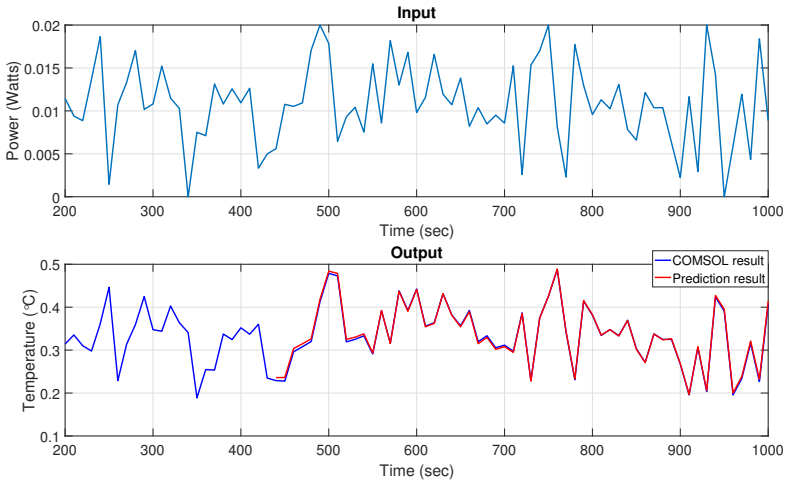


Figure 4.12: Modeling the thermal effect of UEA.

This prediction is then compared with the results obtained from COMSOL. The comparison results are shown in Figure 4.12.

This comparison result indicates that the thermal dynamics of UEA can be captured by the Bayesian RMSPEM method. The prediction performance is 91.0195. The Mean Square Error of the prediction is about 1.2850×10^{-5} °C.

In vitro experiment

The in-vitro thermal effect test vehicle system mentioned before is used to evaluate the prediction accuracy of the simplified thermal model. More specifically, two experiments are conducted. The first experiment randomly generates 2000 PWM signals within the range

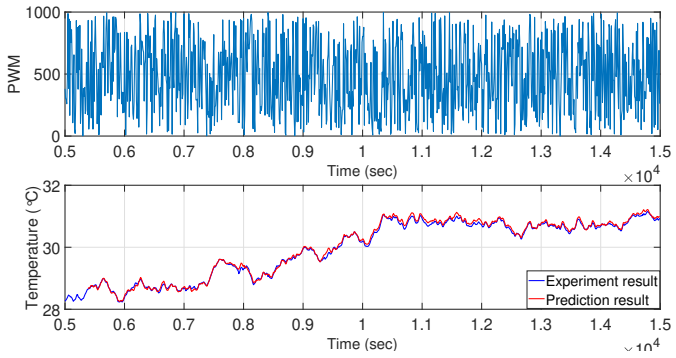
of $[0, 1000]$ using Gaussian distribution and apply the PWM signals to the heat sources on TMTV with a step size of 10 seconds. The temperature recorded by the onboard sensors are then compared with the temperature predicted by the proposed Bayesian RMSPEM method and the prediction error are used for model updating. The Bayesian RMSPEM method implements a 20th order OE model and predicts the temperature measurements of 10 steps ahead. The results are presented in Figure 4.13a. It is demonstrated that the Bayesian RMSPEM accurately predicts the temperature variation despite the varying PWM signal and achieves an overall prediction mean square error of 0.131 °C.

The second experiment generates a random second order low pass filter and applies it to the 2000 random PWM signals. The filtered PWM signal is then applied to the TMTV. This is used to emulate the output of a real thermal management system, where the computed control signal is usually a low frequency signal that depends on various inputs. In this experiment, it is shown in Figure 4.13b that the temperature output can be predicted with a 5th order OE model, which is much simpler than the 20th order OE model used in the first experiment. By taking advantage of this low order model, the computational cost of the proposed method can be greatly reduced. The overall prediction mean square error is about 0.024 °C.

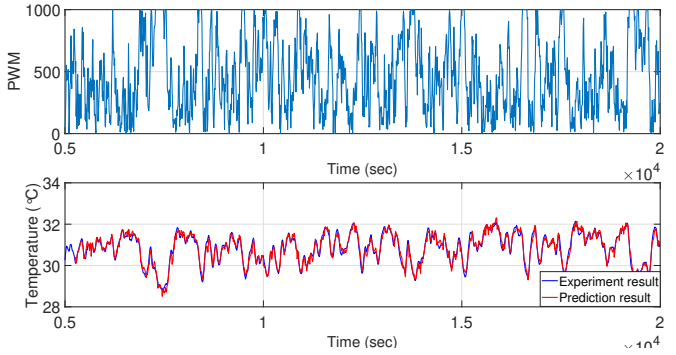
4.8 Adaptive Thermal Management Based on Bayesian Recursive Multiple Step Prediction

The Bayesian recursive multiple step prediction method presented in the previous section enables more accurate prediction with a longer step size, thus can be incorporated into the adaptive thermal management framework mentioned in Section 4.6 to reduce the computational complexity and improve the overall performance. The modified adaptive thermal management framework is tested in this section with an in-vitro experiment.

This experiment is conducted using the in-vitro experiment system in Figure 4.3. The temperature is measured by the onboard sensors with a 1 second rate, the temperature is



(a) Gaussian input.



(b) Filtered Gaussian input.

Figure 4.13: The experiment validation of the Bayesian recursive MSPEM method.

over sampled to improve accuracy with a Kalman filter. The heat sources are controlled via a PWM signal ranging from 0 from 1000. When the PWM signal is 0, the heat sources operate at the maximum state. When the PWM signal is 1000, the heat sources is turned off. The PWM signal is computed by the thermal management method with a 10 seconds step size, which is also the prediction step size. In this experiment, the minimum operating requirement for the heat sources are assumed to be 500. That is, when the computed PWM signal is larger than 500, the PWM sent to the heat sources are set to 1000 and the heat sources are turned off.

Four methods are implemented to control the operating temperature of the test vehicle. The first method is based on the previously published system [41], which turn on the heater when the temperature drops below 21 °C and turn off the heater once the temperature rises above 27 °C. This method is refered to as the benchmark controller in the following context. The second method is what we have proposed in Section 4.6. The prediction model is a simple first order linear model, whose parameters are updated online according to the temperature measurements. This method goes by the name of original controller. The third method implements the MPC based thermal management by using a thermal model predetermined with the Bayesian batch MSPEM, and is therefore named as batch controller. The fourth method controls the operation of test vehicle based on an OE model updated online with Bayesian RMSPEM method. This method is denoted as adaptive method. The optimization problem solved by the original controller, batch controller and adaptive controller can be represented as

$$\begin{aligned} \min \quad & \sum_{t=1}^{k} (P_{max\_uea} - P_{uea})^2 \\ \text{s. t.} \quad & \Delta T(t) = \hat{G}(\theta) P_{uea}(t) \\ & \Delta T(t) \leq \Delta T_{max} \\ & 0 \leq P_{uea} \leq P_{max\_uea} \end{aligned} \tag{4.96}$$

In which, P_{max_uea} represents the power consumption of the maximum working status of the implantable device. $\hat{G}(\theta)$ is the prediction model employed in the MPC with θ being the model parameters. T_{max} is the maximum temperature the implantable device is allowed to operate, which is set to be 27 °C in this experiment.

The results of the experiment are shown in Figure 4.14. The experiment data from 100 seconds to 4000 seconds is used to evaluate the four methods using three metrics. The first metric evaluates how much the safe operation temperature is violated. Since the temperature controlling methods only engage every 10 seconds, the maximum allowable temperature can be violated due to lack of measurement during sampling period and inaccurate prediction model. The metric calculates the average value of temperature overshot (when the temperature is greater than the thermal constraint). This metric can be represented as

$$Violation = \frac{\sum(temperature - 27) \times u(temperature - 27)}{num\ of\ samples}. \tag{4.97}$$

It is shown in Table 4.3 that the adaptive controller has the best performance, while the rest failed to maintain safe operation temperature in many cases. The second metric represents the ratio of time when the implantable device is functional. This corresponds to the time when the heat sources is powered on and the PWM signal sent to the test vehicle is less than 1000. Note that the heat sources are controlled via PWM signal with value ranging from 0 to 1000. The heat sources are turned off when PWM signal is 1000, and turned on when PWM signal is 0. This metric can be calculated via

$$Operation\ ratio = \frac{\sum u(500 - actual\ PWM)}{num\ of\ samples}. \tag{4.98}$$

The third metric reflects the efficiency of the controller, which is calculated as the average ratio of deviation from the maximum power state (when PWM signal is 0). This can be

represented as

$$Operation\ cost = \frac{\sum(\frac{actual\ PWM - maximum\ PWM}{range})^2}{num\ of\ samples}. \tag{4.99}$$

In the equation, $range$ represents the range of PWM signal, which is 1000. This metric assumes that the UEA has best performance when its power consumption reaches maximum. So smaller value indicates better performance. The results in Table 4.3 demonstrates that the adaptive controller has the best performance and the longest operation time. The batch controller has inferior performance due to the gradual deviation of the predetermined prediction model. The original method mitigates model deviation with online model updating, but the generated prediction model is not accurate and leads to poor performance of the MPC. The benchmark controller turns on and off the device according to the temperature, and is therefore the least efficient method.

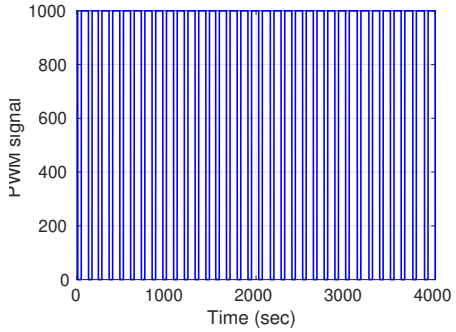
Table 4.3: Results of the comparison experiment

Method	Violation	Operation Ratio	Operation Cost
Benchmark controller	0.1652	0.3274	0.6726
Original controller	0.1081	0.6742	0.4007
Batch controller	0.2575	0.5898	0.4307
Adaptive controller	0.0585	0.7411	0.3402

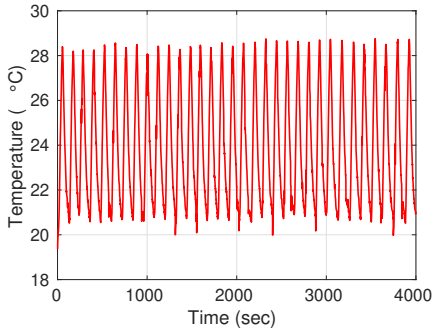
It is shown in this experiment that by dynamically adjusting the working status of the UEA, the operation time can be significantly prolonged. The batch controller and the adaptive controller all have a much longer operation time. Moreover, adaptive controller shows better performance compared to batch controller. This demonstrates the importance of online model updating, as it prevents the prediction model from deviating from the actual system dynamics.

4.9 Summary

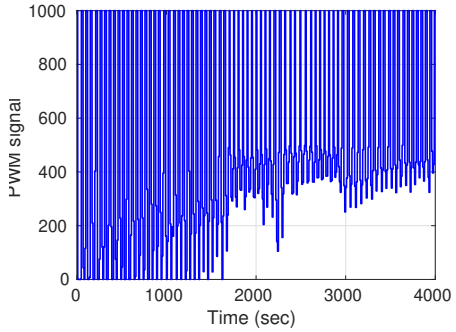
With implantable devices becoming more and more powerful, the heating caused by its operation has drawn growing concern. It is reported that for a neural implant a tempera-



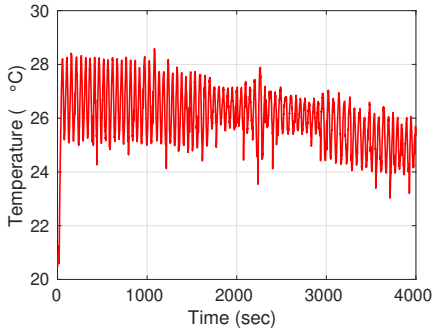
(a) The PWM signal of the benchmark controller.



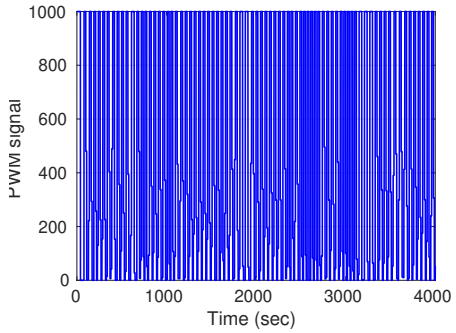
(b) The temperature output of the benchmark controller.



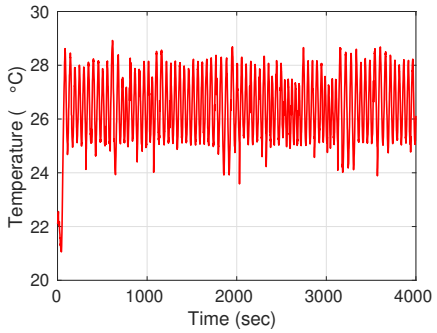
(c) The PWM signal of the original controller.



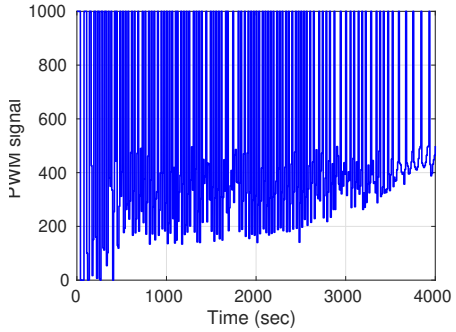
(d) The temperature output of the original controller.



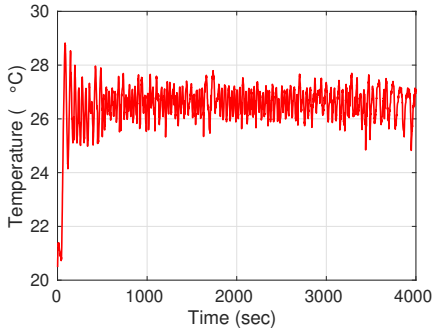
(e) The PWM signal of the batch controller.



(f) The temperature signal of the batch controller.



(g) The PWM signal of the adaptive controller.



(h) The temperature signal of the adaptive controller.

Figure 4.14: The comparison experiment conducted on the in-vitro test vehicle.

ture increase of more than 1 °C could damage the surrounding brain tissue. In this thesis, adaptive thermal management methods are developed to control the heating effect of the implantable device, with a neural prosthesis as an example. The developed method aims to fill in the gap of real time thermal management as opposed to the methods adopted in previous literatures that limit the working status of stimulating electrodes in the design phase. More specifically, two thermal models are proposed to support the real time control and online model parameter estimation methods are developed to update the model parameters based on the real time temperature measurements. These modeling methods are validated with a COMSOL Multiphysics thermal model and an *in vitro* experimental system. Based on the thermal model, the model predictive control is introduced to solve the thermal management problem. Simulation and experiment results indicate that the developed method achieves longer operation time while maintaining safe operating temperature. Furthermore, based on the application requirements, the developed framework can be extended to incorporate the model of energy storage device and an example of joint thermal and power management is demonstrated.

CHAPTER 5
CONCLUSION AND FUTURE WORK

5.1 Dissertation Contributions

This dissertation investigates the power management problem of cyber-physical systems based on the model predictive control method. The modeling of CPS plays an essential role in the development of power management method, as it forms the specifications for the system and reflects the application requirements. The generated models enable predicting system behavior under various input, which makes it possible to optimize system operation and prevent the violation of system constraints in advance.

A model of CPS comprises models of physical processes as well as models of software, computation platforms, and networks. Modeling such a complex systems with reasonable fidelity is challenging, especially when the dynamics of the underlying system is difficult to capture. In this work, the modeling techniques of CPS are demonstrated from two aspects, one is the physics based method and the other is the black box based modeling method. The work presented in this thesis have four major contributions. First, an online state of charge prediction method for supercapacitor is proposed, which updates the device model parameters in a receding horizon fashion, so that the variation of charging/discharging characteristics with operation can be captured. Second, a power management method of maintaining a satisfactory event detection probability for a supercapacitor powered sensor network is proposed. The developed method achieves a constant event detection probability with minimum communication overhead. Third, an online multi-step prediction method is proposed for applications where the system parameter is time varying and the order of the underlying system is hard to determine. The proposed method provides robust prediction performance and has low computational complexity, thus making it suitable for real-time embedded

applications. Fourth, a real-time adaptive thermal management method is proposed for implantable devices to optimize their operation while maintaining a safe operating temperature. The proposed method is evaluated through COMSOL simulation and in-vitro experiment. The afore mentioned contributions are also explained with more details in the following section.

5.2 Conclusions

Chapter 2 summarizes the research work that are related to the subject of this thesis. The special property of cyber-physical system and the challenge it brought to the power management are first explained. The model predictive control is then introduced to solve the power management problem of two cyber-physical systems: a supercapacitor powered wireless sensor network and an implantable neural prothesis. These two systems demonstrate two major modeling methods that are employed in MPC: the physics-based modeling method and the black box based modeling method. In the supercapacitor powered wireless sensor network system, the dynamics of supercapacitor is difficult to model using traditional methods. An online state of charge prediction method is proposed based on the supercapacitor physics to account for the different behavior of supercapacitor under varying load. In the neural prothesis system, the thermal effect of implantable device is very difficult to model due to different device dimension and the complex environment where the device is implanted. Therefore, an online multi-step prediction method is proposed, which is able to generate an accurate temperature prediction within a horizon despite the noise and varying environment.

Based on the supercapacitor online state of charge prediction method, Chapter 3 proposes the adaptive power management method for supercapacitor powered radar sensor network. The proposed method start by decomposing the network event detection probability to the node level as the quality of service. Then a power management method based on the model predictive control and particle swarm optimization is proposed for track-

ing the reference quality service of each node while satisfying the operation constraints. The effectiveness of the proposed method is demonstrated through three simulation studies which cover both single node and network scenarios.

Chapter 4 proposes a real-time adaptive thermal management method for implantable devices like neural prosthesis based on the online multi-step prediction method. The thermal effect of the device is predicted within a horizon and a model predictive control based method is developed to optimize the device operation while maintaining a safe temperature. The performance of the proposed method is evaluated via both simulation studies and in-vitro experiments and the result indicate that the method achieves longer operation time and better overall performance while maintaining the safe operation temperature.

5.3 Future Work

In this dissertation, we investigate the power management of implantable devices, with a focus on applications that have a dominant heat source. In many real applications, the circuit spatial distriubiton of some implantable devices and the heating effect of electromagnetic field generated by wireless power transfer and data communication could make the power managment problem more complicated. To solve the problem when multiple heat sources are present, the power management framework developed in this paper will be extended to predict and control the thermal dynamics of different heat sources simulataneously. Moreover, a simple thermal constraint is employed in the current framework to prevent the temperature from reaching to the dangerous level. However, the impact of overheating to the surrounding tissues can be more complicated. This effect needs to be modeled and incorporated into the power management framework to optimize the application performance while maintaining safe operation.

REFERENCES

[1] Stamatis Karnouskos. “Cyber-physical systems in the smartgrid”. In: *2011 9th IEEE International Conference on Industrial Informatics (INDIN)*. IEEE, 2011, pp. 20–23.

[2] Vijay Kumar Jonathan Fink Alejandro Ribeiro. “Robust control for mobility and wireless communication in cyber-physical systems with application to robot teams”. In: *Proceedings of the IEEE*. Vol. 100. 1. 2012, pp. 164–178.

[3] Ovunc Kocabas, Tolga Soyata, and Mehmet K. Aktas. “Emerging Security Mechanisms for Medical Cyber Physical Systems”. In: *IEEE/ACM Transactions on Computational Biology and Bioinformatics* 13.3 (2016), pp. 401–416.

[4] Yunbo Wang, Mehmet C. Vuran, and Steve Goddard. “Cyber-physical systems in industrial process control”. In: *ACM SIGBED Review* 5.1 (2008), pp. 1–2.

[5] Yu-Geng XI, De-Wei LI, and Shu LIN. “Model Predictive Control-Status and Challenges”. In: *Acta Automatica Sinica* 39.3 (2014), pp. 222–236.

[6] Tim Salsbury, Prashant Mhaskar, and S. Joe Qin. “Predictive control methods to improve energy efficiency and reduce demand in buildings”. In: *Computers & Chemical Engineering* 51 (2013), pp. 77–85.

[7] Patrick Bouffard, Anil Aswani, and Claire Tomlin. “Learning-based model predictive control on a quadrotor: Onboard implementation and experimental results”. In: *2012 IEEE International Conference on Robotics and Automation*. IEEE, 2012.

[8] Enrico Silani and Marco Lovera. “Magnetic spacecraft attitude control: a survey and some new results”. In: *Control Engineering Practice* 13.3 (2005), pp. 357–371.

[9] Wenlin Wang, Daniel E. Rivera, and Karl G. Kempf. “Model predictive control strategies for supply chain management in semiconductor manufacturing”. In: *International Journal of Production Economics* 107.1 (2007), pp. 56–77.

[10] F. Borrelli et al. “MPC-based approach to active steering for autonomous vehicle systems”. In: *International Journal of Vehicle Autonomous Systems* 3.2/3/4 (2005), p. 265.

[11] Torsten Söderström Lennart Ljung. *Theory and practice of recursive identification*. The MIT Press, 1983.

[12] Lennart Ljung. *System Identification: Theory For the User*. 2 edition. Upper Saddle River, N.J. USA: Prentice Hall, 1999.

[13] Jay H. Lee Manfred Morari. "Model predictive control: past, present and future". In: *Computers & Chemical Engineering* 23.4 (1999), pp. 667–682.

[14] Yogesh Sankarasubramaniam Erdal Cayirci Ian F. Akyildiz Weilian Su. "A survey on sensor networks". In: *IEEE Communications magazine* 40.8 (2002), pp. 102–114.

[15] Dipak Ghosal Jennifer Yick Biswanath Mukherjee. "Wireless sensor network survey". In: *Computer networks* 52.12 (2008), pp. 2292–2330.

[16] P. Dutta et al. "Design of a wireless sensor network platform for detecting rare, random, and ephemeral events". In: *IPSN 2005. Fourth International Symposium on Information Processing in Sensor Networks, 2005.* IEEE, 2005.

[17] Jonathan W. Kimball, Brian T. Kuhn, and Robert S. Balog. "A System Design Approach for Unattended Solar Energy Harvesting Supply". In: *IEEE Transactions on Power Electronics* 24.4 (2009), pp. 952–962.

[18] Fabio Ongaro, Stefano Saggini, and Paolo Mattavelli. "Li-Ion Battery-Supercapacitor Hybrid Storage System for a Long Lifetime, Photovoltaic-Based Wireless Sensor Network". In: *IEEE Transactions on Power Electronics* 27.9 (2012), pp. 3944–3952.

[19] Farhan Simjee and Pai H. Chou. "Everlast: long-life, supercapacitor-operated wireless sensor node". In: *Proceedings of the 2006 international symposium on Low power electronics and design*. ACM Press, 2006, pp. 197–202.

[20] R. Kötz and M. Carlen. "Principles and applications of electrochemical capacitors". In: *Electrochimica Acta* 45.15-16 (2000), pp. 2483–2498.

[21] Tian He Zhi-Li Zhang. Ting Zhu Ziguo Zhong. "Energy-synchronized computing for sustainable sensor networks". In: *Ad Hoc Networks* (2010).

[22] Hengzhao Yang and Ying Zhang. "Analysis of Supercapacitor Energy Loss for Power Management in Environmentally Powered Wireless Sensor Nodes". In: *IEEE Transactions on Power Electronics* 28.11 (2013), pp. 5391–5403.

[23] Ming Li et al. "Multi-user data sharing in radar sensor networks". In: *Proceedings of the 5th international conference on Embedded networked sensor systems - SenSys '07*. ACM Press, 2007.

[24] Hassan Jaleel, Amir Rahmani, and Magnus Egerstedt. "Duty cycle scheduling in dynamic sensor networks for controlling event detection probabilities". In: *Proceedings of the 2011 American Control Conference*. IEEE, 2011.

[25] Dimitrios D. Vergados Nikolaos A. Pantazis. “A survey on power control issues in wireless sensor networks”. In: *IEEE Communications Surveys & Tutorials* 9.4 (2007), pp. 86–107.

[26] Mario Di Francesco-Andrea Passarella. Giuseppe Anastasi Marco Conti. “Energy conservation in wireless sensor networks: A survey”. In: *Ad hoc networks* 7.3 (2009), pp. 537–568.

[27] Hassan Jaleel, Amir Rahmani, and Magnus Egerstedt. “Probabilistic Lifetime Maximization of Sensor Networks”. In: *IEEE Transactions on Automatic Control* 58.2 (2013), pp. 534–539.

[28] Elias Greenbaum and David Zhou, eds. *Implantable Neural Prostheses*. New York: Springer US, 2009.

[29] Nadey S. Hakim. *Artificial Organs*. London: Springer, 2009.

[30] K.D. Wise et al. “Microelectrodes, Microelectronics, and Implantable Neural Microsystems”. In: *Proceedings of the IEEE* 96.7 (2008), pp. 1184–1202.

[31] Alireza Rajabi-Tavakkol, Amir M Sodagar, and Mohammad H Refan. “New architecture for wireless implantable neural recording microsystems based on frequency-division multiplexing”. In: *2010 Annual International Conference of the IEEE Engineering in Medicine and Biology*. IEEE, 2010.

[32] P S Ruggera et al. “In vitroassessment of tissue heating near metallic medical implants by exposure to pulsed radio frequency diathermy”. In: *Physics in Medicine and Biology* 48.17 (2003), pp. 2919–2928.

[33] J. Dommerholt, T. Issa, and J. D. Nutt. “DBS and diathermy interaction induces severe CNS damage”. In: *Neurology* 57.12 (2001), pp. 2324–2325.

[34] Gerald M. Saidel Charles R. Davies Timothy M. Seese Hiroaki Harasaki. “Characterization of tissue morphology, angiogenesis, and temperature in the adaptive response of muscle tissue to chronic heating”. In: *Laboratory investigation; a journal of technical methods and pathology* 78.12 (1998), pp. 1553–1562.

[35] M. Ueda, J. Bure, and J. Fischer. “Spreading depression elicited by thermal effects of ultrasonic irradiation of cerebral cortex in rats”. In: *Journal of Neurobiology* 8.4 (1977), pp. 381–393.

[36] Takanori Fujii and Yasuhiko Ibata. “Effects of heating on electrical activities of guinea pig olfactory cortical slices”. In: *Pflügers Archiv European Journal of Physiology* 392.3 (1982), pp. 257–260.

[37] Joseph C. LaManna et al. “Stimulus-activated changes in brain tissue temperature in the anesthetized rat”. In: *Metabolic Brain Disease* 4.4 (1989), pp. 225–237.

[38] H. H. Pennes. “Analysis of Tissue and Arterial Blood Temperatures in the Resting Human Forearm”. In: *Journal of Applied Physiology* 85.1 (1998), pp. 5–34.

[39] G. Lazzi. “Thermal effects of bioimplants”. In: *IEEE Engineering in Medicine and Biology Magazine* 24.5 (2005), pp. 75–81.

[40] Sohee Kim et al. “Thermal Impact of an Active 3-D Microelectrode Array Implanted in the Brain”. In: *IEEE Transactions on Neural Systems and Rehabilitation Engineering* 15.4 (2007), pp. 493–501.

[41] Christian T Wentz et al. “A wirelessly powered and controlled device for optical neural control of freely-behaving animals”. In: *Journal of Neural Engineering* 8.4 (2011), p. 046021.

[42] B. D. Ripley and P. Hall. “Introduction to the Theory of Coverage Processes.” In: *Biometrics* 45.3 (1989), p. 1034.

[43] H. Ying R. Chai and Y. Zhang. “Supercapacitor charge redistribution analysis for power management of wireless sensor networks”. In: *IET Power Electronics* 10.2 (2017), pp. 169–177.

[44] R. Galvan-Guerra P. Martin and M. Egerstedt. “Power-Aware Sensor Coverage: An Optimal Control Approach”. In: *19th International Symposium on Mathematical Theory of Networks and Systems (MTNS)*. 2010.

[45] C. Hsin and M. Liu. “Network coverage using low duty-cycled sensors: random & coordinated sleep algorithms”. In: *Proceedings of the 3rd international symposium on Information processing in sensor networks*. 2004, pp. 433–442.

[46] P. Chen A. Marshall S. Yang Y. Tahir and J. McCann. “Distributed optimization in energy harvesting sensor networks with dynamic in-network data processing”. In: *The 35th Annual IEEE International Conference on Computer Communications (INFOCOM)*. 2016, pp. 1–9.

[47] Andrew Burke. “Ultracapacitors: why, how, and where is the technology”. In: *Journal of Power Sources* 91.1 (2000), pp. 37–50.

[48] Maximilian Kaus, Julia Kowal, and Dirk Uwe Sauer. “Modelling the effects of charge redistribution during self-discharge of supercapacitors”. In: *Electrochimica Acta* 55.25 (2010), pp. 7516–7523.

[49] Ronan German et al. “Improved Supercapacitor Floating Ageing Interpretation Through Multipore Impedance Model Parameters Evolution”. In: *IEEE Transactions on Power Electronics* 29.7 (2014), pp. 3669–3678.

[50] E.-H. El Brouji et al. “Impact of Calendar Life and Cycling Ageing on Supercapacitor Performance”. In: *IEEE Transactions on Vehicular Technology* 58.8 (2009), pp. 3917–3929.

[51] Ying Zhang and Hengzhao Yang. “Modeling and characterization of supercapacitors for wireless sensor network applications”. In: *Journal of Power Sources* 196.8 (2011), pp. 4128–4135.

[52] J. Polastre, R. Szewczyk, and D. Culler. “Telos: enabling ultra-low power wireless research”. In: *IPSN 2005. Fourth International Symposium on Information Processing in Sensor Networks, 2005.* IEEE.

[53] D. Brunelli et al. “Design of a Solar-Harvesting Circuit for Batteryless Embedded Systems”. In: *IEEE Transactions on Circuits and Systems I: Regular Papers* 56.11 (2009), pp. 2519–2528.

[54] Hengzhao Yang and Ying Zhang. “Self-discharge analysis and characterization of supercapacitors for environmentally powered wireless sensor network applications”. In: *Journal of Power Sources* 196.20 (2011), pp. 8866–8873.

[55] J.C. Butcher. *Numerical Methods for Ordinary Differential Equations*. John Wiley & Sons, Ltd, 2003.

[56] W.H. Press. *Numerical Recipes: The Art of Scientific Computing (3rd ed.)* Cambridge university press, 2007.

[57] Ruizhi Chai, Ying Zhang, and Maysam Ghovanloo. “Joint power and thermal management for implantable devices”. In: *2015 IEEE Biomedical Circuits and Systems Conference (BioCAS)*. IEEE, 2015.

[58] *Maccor Battery and Cell Test Equipment (Model 4300). Available: http://www.maccor.com/Products*

[59] X. Gong F. Xu H. Chen and Q. Mei. “Fast nonlinear model predictive control on FPGA using particle swarm optimization”. In: *IEEE Transactions on Industrial Electronics* 63.1 (2016), pp. 310–321.

[60] A. Andreas C. Maxey. *NREL Report No. DA-5500-56512*. Tech. rep. Oak Ridge National Laboratory (ORNL), 2007.

[61] D. Atienza J.R. Piorno C. Bergonzini and T.S. Rosing. “Prediction and management in energy harvested wireless sensor nodes”. In: *1st International Conference on*

Wireless Communication, Vehicular Technology, Information Theory and Aerospace & Electronic Systems Technology. 2009, pp. 6–10.

[62] Evgeny Katz. "Implantable Bioelectronics - Editorial Introduction". In: *Implantable Bioelectronics*. Wiley-VCH Verlag GmbH & Co. KGaA, 2014, pp. 1–5.

[63] N. L. Opie et al. "Heating of the Eye by a Retinal Prosthesis: Modeling, Cadaver and In Vivo Study". In: *IEEE Transactions on Biomedical Engineering* 59.2 (2012), pp. 339–345.

[64] Kanber Mithat Silay, Catherine Dehollain, and Michel Declercq. "Numerical Thermal Analysis of a Wireless Cortical Implant with Two-Body Packaging". In: *Bio-NanoScience* 1.3 (2011), pp. 78–88.

[65] C. H. W. Oey and S. Moh. "A survey on temperature-aware routing protocols in wireless body sensor networks". In: *Sensors* 13.8 (2013), pp. 9860–9877.

[66] Rossi Kamal and Choong Seon Hong. "Fault tolerant virtual backbone for minimum temperature in in vivo sensor networks". In: *2012 IEEE International Conference on Communications (ICC)*. IEEE, 2012.

[67] Muhammad Monowar and Fuad Bajaber. "On Designing Thermal-Aware Localized QoS Routing Protocol for in-vivo Sensor Nodes in Wireless Body Area Networks". In: *Sensors* 15.6 (2015), pp. 14016–14044.

[68] Q. Tang et al. "Communication Scheduling to Minimize Thermal Effects of Implanted Biosensor Networks in Homogeneous Tissue". In: *IEEE Transactions on Biomedical Engineering* 52.7 (2005), pp. 1285–1294.

[69] Yi Luo, Masoud Dahmardeh, and Kenichi Takahata. "Biocompatible circuit-breaker chip for thermal management of biomedical microsystems". In: *Journal of Micromechanics and Microengineering* 25.5 (2015), p. 055002.

[70] C.M. Krishna and I. Koren. "Thermal-aware management techniques for cyber-physical systems". In: *Sustainable Computing: Informatics and Systems* 15 (2017), pp. 39–51.

[71] W. Liu J. D. Weiland S. C. DeMarco G. Lazzi and M. S. Humayun. "Computed SAR and thermal elevation in a 0.25-mm 2-D model of the human eye and head in response to an implanted retinal stimulator-Part I: Models and methods". In: *IEEE Transactions on Antennas and Propagation* 51.9 (2003), pp. 2274–2285.

[72] Jean-Louis Dillenseger and Simon Esneault. "Fast FFT-based bioheat transfer equation computation". In: *Computers in Biology and Medicine* 40.2 (2010), pp. 119–123.

[73] G. Carluccio et al. “An Approach to Rapid Calculation of Temperature Change in Tissue Using Spatial Filters to Approximate Effects of Thermal Conduction”. In: *IEEE Transactions on Biomedical Engineering* 60.6 (2013), pp. 1735–1741.

[74] Vinit Singh et al. “On the Thermal Elevation of a 60-Electrode Epiretinal Prosthesis for the Blind”. In: *IEEE Transactions on Biomedical Circuits and Systems* 2.4 (2008), pp. 289–300.

[75] Shiva K. Das, Scott T. Clegg, and Thaddeus V. Samulski. “Computational techniques for fast hyperthermia temperature optimization”. In: *Medical Physics* 26.2 (1999), pp. 319–328.

[76] M. Pedram and S. Nazarian. “Thermal Modeling, Analysis, and Management in VLSI Circuits: Principles and Methods”. In: *Proceedings of the IEEE* 94.8 (2006), pp. 1487–1501.

[77] Geng Sun Hongsheng Li Ruizhi Chai Ying Zhang. “Power Management for Controlling Event Detection Probability of Supercapacitor Powered Sensor Networks”. In: *2018 American Control Conference*. (accepted). 2018.

[78] Rodrigo Alvite, Alain Segundo, and Claudio Garci. “Model Predictive Control Relevant Identification”. In: *Frontiers in Advanced Control Systems*. InTech, 2012.

[79] M. Farina and L. Piroddi. “Simulation error minimization identification based on multi-stage prediction”. In: *International Journal of Adaptive Control and Signal Processing* 25.5 (2011), pp. 389–406.

[80] Sébastien Ethier and Mohamad Sawan. “Exponential Current Pulse Generation for Efficient Very High-Impedance Multisite Stimulation”. In: *IEEE Transactions on Biomedical Circuits and Systems* 5.1 (2011), pp. 30–38.

[81] Fumin Zhang, Zhenwu Shi, and Wayne Wolf. “A Dynamic Battery Model for Co-design in Cyber-Physical Systems”. In: *2009 29th IEEE International Conference on Distributed Computing Systems Workshops*. IEEE, 2009.

[83] M. Grant and S. Boyd. *CVX: Matlab software for disciplined convex programming, version 2.0 beta (http://cvxr.com/cvx)*. Tech. rep. 2013.

[84] O. Nelles. *Nonlinear system identification: from classical approaches to neural networks and fuzzy models*. Springer Science & Business Media, 2013.

[85] G. Pillonetto et al. “Kernel methods in system identification, machine learning and function estimation: A survey”. In: *Automatica* 50.3 (2014), pp. 657–682.

[86] G. Pillonetto et al. “Regularized linear system identification using atomic, nuclear and kernel-based norms: the role of the stability constraint”. In: *Automatica* 69 (2016), pp. 137–149.

[87] M. Zorzi and A. Chiuso. “The harmonic analysis of kernel functions”. In: *Automatica* 94 (2018), pp. 125–137.

[88] M. Zorzi and A. Chiuso. “Sparse plus low rank network identification: A nonparametric approach”. In: *Automatica* 76 (2017), pp. 355–366.

[89] P. C. Young. *Recursive estimation and time-series analysis: an introduction*. Springer Science & Business Media, 2012.

[90] G. Prando, D. Romeres, and A. Chiuso. “Online identification of time-varying systems: A Bayesian approach”. In: *2016 IEEE 55th Conference on Decision and Control (CDC)*. Vol. 1. 2016.